

68

\$2.95USA

Australia
Singapore
Malaysia

A \$ 4.75
S \$ 9.45
M \$ 9.45

New Zealand
Hong Kong
Sweden

NZ \$ 6.50
H \$23.50
30:-SEK

MICRO JOURNAL

VOLUME V ISSUE IX • Devoted to the 68XX User • September 1983
"Small Computers Doing Big Things"

SERVING THE 68XX USER WORLDWIDE





FREE FROM LIMITATIONS. A TEACHER CAN UNLOCK IMAGINATIONS.

By 1985, fully 75% of all jobs will involve computers. So for today's teachers, selecting the right computer is an awesome responsibility. It's a decision you can feel confident about with Southwest Technical Products Corporation's S+ computer system. The S+ is specially designed for a student's tomorrow and a teacher's today. It's built tough. It's simple to operate. You can teach without becoming a hardware specialist.

The S+ is fluent in a variety of languages: BASIC, PASCAL, COBOL, FORTRAN, PILOT,

FORTH, "C," and MUMPS and allows you simultaneous use in the classroom. This flexibility enables you to use your own curriculum, not a machine's version. The system also allows for individual learning styles. Students go as fast and as far as the teacher chooses.

Even though it's priced like a smaller personal computer, the S+ has the multi-task power and expansion abilities of a professional computer. It can grow with you and your students.

Tomorrow is in your hands.

today. Call Fisher Scientific, exclusive distributor of our S+ system, and you'll get the key to unlock imaginations.

For more information. Call toll-free 800-621-4769. Illinois customers call collect (312) 378-7770.

Fisher Scientific

An  **ALLIED** Company

Educational Materials Division
4901 W. LeMoyné St.,
Chicago, Illinois 60651

The UniFLEX™ Operating System extracts



every last drop

**from the 8 bit 6809 microprocessor allowing it to
outperform many 16 bit systems**

With the UniFLEX™ Operating System, the 8 bit 6809 microprocessor can perform as well as larger CPUs in a multi-user, multi-tasking environment.

Independently developed from the ground up, UniFLEX™ closely models the features found in the UNIX™ Operating System. And in two years of use, UniFLEX™ has proven the abilities of the 6809 to perform large system functions when incorporated into a properly designed mainframe.

Some of the features supported include:

- full multi-user, multi-tasking capabilities
- hierarchical file systems
- device independent I/O
- four Gigabyte disk capacities
- full file protection
- inter-task communication via pipes
- I/O redirection
- task swapping for efficient memory usage
- full random-access files
- comprehensive shell command language
- foreground-background jobs
- electronic mail and printer spooling
- system accounting facilities

The support software currently available for use under UniFLEX™ is extensive. A sampling of the programs available includes:

- native C compiler (full implementation)
- native Pascal compiler
- FORTRAN 77 ANSI Subset compiler
- COBOL compiler with ISAM files, Report Writer & Sort/Merge
- Extended BASIC interpreter
- Extended BASIC precompiler
- text editing and processing software
- enhanced printer spooler
- variety of absolute and relocatable assemblers
- debug and diagnostic packages

Technical Systems Consultants, Inc. also offers a line of single user FLEX™ software products for 6800 and 6809 processors. For those having an absolute need for a 16 bit processor, UniFLEX™ will be available through OEM licensing arrangements for the 68000 microprocessor. Please call or write for additional information on individual products or OEM licensing arrangements.

UNIX™ is a trademark of Bell Laboratories.
FLEX™ and UniFLEX™ are trademarks of Technical Systems Consultants, Inc.

 **technical systems
consultants, inc.**

111 Providence Road
Chapel Hill, North Carolina 27514
(919) 493-1451

'68'

Portions of the text for 68 MICRO JOURNAL was prepared using the following furnished hard/software.

COMPUTERS-HARDWARE

Southwest Technical Products
219 W. Rhapsody
San Antonio, Texas 78216
S09-5/8 DMF disk-COSI-8212W-Sprint 3 Printer

GIMIX Inc.
1337 West 37th Place
Chicago, IL 60609
Super Mainframe-OS9-FLEX-Assorted Hardware

EDITORS-WORD PROCESSORS

Technical Systems Consultants, Inc.
111 Providence Road
Chapel Hill, NC 27514
FLEX-Editor-Processor

Great Plains Computer Company, Inc.
PO Box 916
Idaho Falls, ID 83401
STYLO-Mail Merge

Editorial Staff

Don Williams Sr.	Publisher
Larry E. Williams	Executive Editor
Tom E. Williams	Production Editor
Robert (Bob) Nay	Color Editor

Administrative Staff

Mary Robertson	Office Manager
Joyce Williams	Accounting
Carolyn Williams	Subscriptions
Penny Williams	File Management

Contributing Editors

Ron Anderson
N. m. Commo
Peter Dibble
Dr. Theo Elbert
William E. Fisher
Or. E.M. Pass

Special Technical Projects

Clay Abrams K6AEP
Tom Hunt

CONTENTS

Vol. V, Issue IX

SEPTEMBER '83

FLEX USER NOTES.....	10	Anderson
COLOR USER NOTES.....	12	Nay
OS9 USER NOTES.....	12	Dibble
STASM ASSEMBLER..Review.....	15	Anderson
SELECTIVE DIR.CMD.....	16	Gitelson
XADR.....Review.....	20	Staff
CONDUIT...OS9...Review.....	20	Pass
PURCHASE ORDER WRITER.....	23	Cummings
EXTENDED MEMORY FOR OLD SYS....	26	Scudlere
MP-C to MP-T for SWTPC.....	26	Wright
WHIMSICAL..A Compiler..Review..	27	Anderson
TRANSPORT WITH DELIGHT.....	29	Fowler
GSPL..A Compiler..Review.....	31	Anderson
BIT BUCKET.....	33	All of us
A SING E BOARD 6809 COMPUTER...	38	Gitelson
		Sanseska Sys.
Classified.....	50	

MICRO JOURNAL

Send All Correspondence To:

Computer Publishing Center
68 MICRO JOURNAL
5900 Cassandra Smith
PO Box 849
Hixson, TN 37343
615 842-4600

Copyrighted 1983 by Computer Publishing, Inc. (CPI)

68' Micro Journal is published 12 times a year by Computer Publishing Inc. Second Class Postage Paid ISSN 0194-5025 at Hixson, Tenn. and additional entries. Postmaster: send Form 3579 to 68' Micro Journal, PO Box 849, Hixson, Tennessee.

SUBSCRIPTION RATES

USA

1-Year \$24.50 2-Years \$42.50 3-Years \$64.50

FOREIGN

See Page 52

Items Submitted for Publication

Articles submitted for publication should be accompanied by the authors full name, address, date and telephone number. It is preferred that articles be submitted on either 5 or 8 inch diskette in TSC Editor format or STYLO format. All diskettes will be returned.

The following TSC Text Processor commands ONLY should be used (due to our proportional processor): .sp space, .pp paragraph, .fl fill and .nf no fill. Also please do not format within the text with multiple spaces. The rest we will enter at time of editing.

STYLO commands are all acceptable except the .pg page command, we print edited text files in continuous text.

All articles submitted on diskettes should be in TSC FLEX[™] format, either FLEX2 6800, or FLEX9 6809 any version.

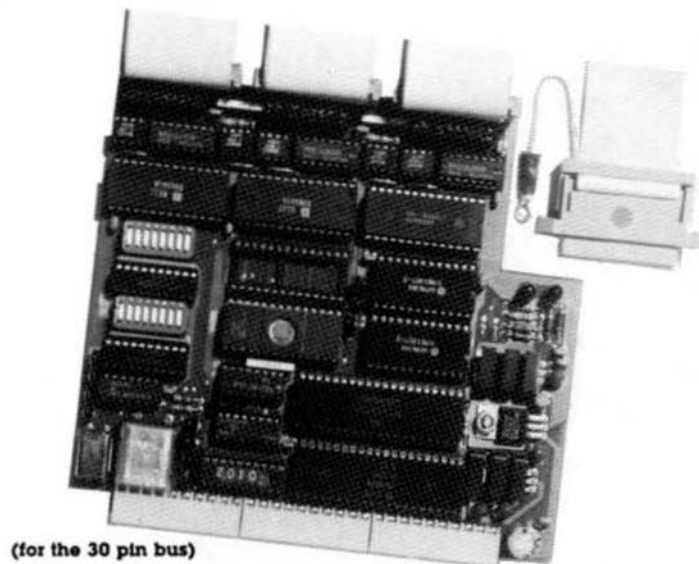
If articles are submitted on paper they should be on white 8X11 bond or better grade paper. No hand written articles (hand written or drawn art accepted). All paper submitted articles will be photo reproduced. This requires that they be typed or produced with a dark ribbon (no blue), single spaced and type font no smaller than 'elite' or 12 pitch. Typed text should be approximately 7 inches wide (will be reduced to column width of 3 1/2 inches). Please use a dark ribbon!

All letters to the editor should also comply with the above and bear a signature. Letters of 'gripes' as well as 'praise' are solicited. We attempt to publish all letters to the editor verbatim, however, we reserve the right to reject any submission for lack of 'good taste'. We reserve the right to define what constitutes 'good taste'.

Advertising: Commercial advertisers please contact the 68 Micro Journal advertising department for current rate sheet and requirements.

Classified: All classified must be non-commercial. Maximum 20 words per classified ad. Those consisting of more than 20 words should be figured at .35 cents per word. 20 words or less \$7.50 minimum, one time, paid in advance. No classified ads accepted by telephone.

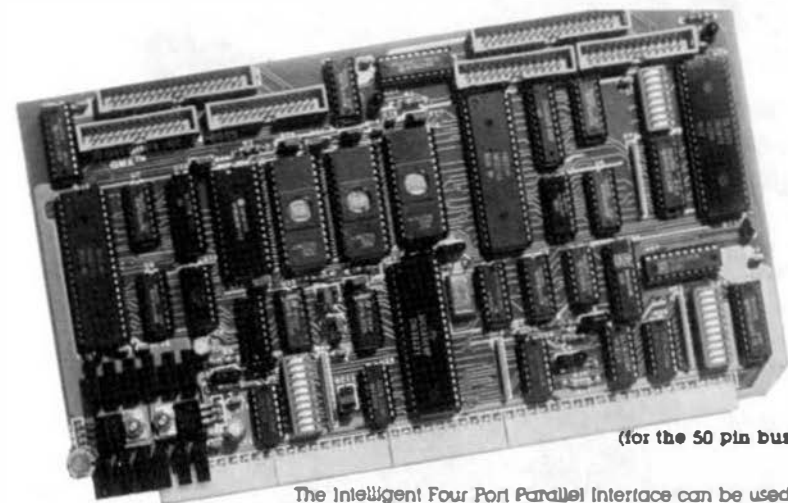
Intelligent Serial I/O Processor Board #11



(for the 30 pin bus)



Intelligent Parallel I/O Processor Board #12



(for the 50 pin bus)

The GIMIX Intelligent Three-port RS-232C Serial interface can significantly increase the throughput of a multi-user system. By buffering data transfers between users and the system, and preprocessing the data, it reduces the number of interrupts to the host CPU, allowing the host more time for other processing tasks.

Features:

- Independent on-board 2MHz 68B09 CPU
- Up to 20K of on-board memory (EPROM and RAM)
- Buffered data transfer between host and on-board CPUs using a Z8038 FIO with 128 byte bi-directional FIFO buffer and mailbox message capabilities
- Three RS-232C serial I/O ports (6551As) with software selectable baud rates, word length, stop bits, and parity.
- Each port has five "handshake" lines for modem control applications.
- The on-board 6809 can be reset by the host processor
- Compatible with memory-to-memory DMA transfers to/from the GMX 6809 CPU III.
- Sense switches and status LEDs that can be used to select software options and indicate board status
- Separate 26-pin cable connections for each port

Appropriate on-board firmware and operating system drivers are required. Uses up to three #95 cable sets (DB-25S connectors).

OS-9 firmware and drivers for the Intelligent 3-port Serial Interface

The OS-9 firmware and drivers enhance the performance of multi-user systems, while providing functions equivalent to the standard ACIA type drivers normally used for terminals and serial printers. Input line editing functions (backspace, echo, line dup and repeat, etc.) are handled by the I/O board, rather than the host, allowing the host more time for other processing tasks. The host is only interrupted when a complete input line (terminated by a "CR") is entered, or certain special characters are received. Input and output data are buffered on the I/O board so that the host can perform other tasks while serial data is being transmitted or received. When used with the GMX CPU III, block data transfers between the I/O board and the host use the CPU's memory-to-memory DMA to further enhance throughput. In addition to performance enhancements, features such as software selectable baud rates and transmission characteristics (number of data bits, stop bits, parity, etc.) are provided. The board also transmits "messages" to any or all I/O ports to indicate that the I/O interface is ready and "waiting for the host", and that the host is "on-line" and has opened a path to the port. Messages to individual ports can be disabled.


The Intelligent Four Port Parallel Interface can be used to improve system performance by buffering data transfers to parallel peripherals such as printers and/or by buffering and pre-processing parallel input data from keyboards, sensors, etc.

Features:

- Independent on-board 2MHz 68B09 CPU
- Up to 32K of on-board memory (EPROM and RAM)
- Buffered data transfer between host and on-board CPUs using a Z8038 FIO with 128 byte bi-directional FIFO buffer and mailbox message capabilities.
- Four fully buffered 8-bit parallel ports with handshaking and input/output latches (two 6522 VIAs). Each 6522 also has two 16-bit counters and a shift register for serial data transfers.
- Software programmable direction for each bit on two of the four ports (1 per VIA), the other 2 ports can be individually programmed as 8 in or 8 out. The bi-directional handshake lines can be programmed as inputs or outputs.
- The on-board 6809 can be reset by the host processor
- Full 20-bit address decoding; it can be addressed on any 4 byte boundary in 1M byte of address space.
- Compatible with memory-to-memory DMA transfers to/from the GMX 6809 CPU III.

Appropriate on-board firmware and operating system drivers are required. Uses up to four #95 cable sets (DB-25P connectors) or two 36-pin cable sets with Centronics compatible connectors. Centronics compatible cable sets include a back panel connector plate for the Classy Chassis. Back panel to printer cables are also available.

THE COMPLETE 6809



The Microware C
is a fully implemented
Kernighan and Ritchie
standard C Compiler
with relocatable
assembler, linker, profiler,
UNIX* and OS-9*
standard libraries.

Suggested U.S.
retail price \$400.

**Now available
from**



MICROWARE®
5835 Grand Avenue
Des Moines, Iowa 50312
515-279-8844
Telex: 910-520-2535

*UNIX is a trademark of Bell Laboratories. OS-9 is a trademark of Microware and Motorola, Inc.

FLEX™ USER NOTES

THE 6800-6809 BOOK

By: Ronald W. Anderson

As published in 68 MICRO JOURNAL™

The publishers of 68 MICRO JOURNAL are proud to announce the publication of Ron Anderson's **FLEX USER NOTES**, in book form. This popular monthly column has been a regular feature in 68 MICRO JOURNAL SINCE 1979. It has earned the respect of thousands of 68 MICRO JOURNAL readers over the years. In fact, Ron's column has been described as the 'Bible' for 68XX users, by some of the world's leading microprocessor professionals. Now all his columns are being published, in whole, as the most needed and popular 68XX book available. Over the years Ron's column has been one of the most popular in 68 MICRO JOURNAL. And of course 68 MICRO JOURNAL is the most popular 68XX magazine published.

As a **SPECIAL BONUS** all the source listing in the book will be available on disk for the low price of: **FLEX™** format only — 5" \$12.95 — 8" \$16.95 plus \$2.50 shipping and handling, if ordered with the book. If ordered separately the price of the disks will be: 5" \$17.95 — 8" \$19.95 plus \$2.50 shipping and handling.

Listed below are a few of the **TEXT** files included in the book and on diskette.

All **TEXT** files in the book are on the disks.

LOGO.C1	File load program to offset memory — ASM PIC
MOVE.C1	Memory move program — ASM PIC
DUMP.C1	Printer dump program — uses LOGO — ASM PIC
SUBTEST.C1	Simulation of 6800 code to 6809, show differences — ASM
TERMEM.C2	Modem input to disk (or other port input to disk) — ASM
M.C2	Output a file to modem (or another port) — ASM
PRINT.C3	Parallel (enhanced) printer driver — ASM
MODEM.C2	TTL output to CRT and modem (or other port) — ASM
SCIPKG.C1	Scientific math routines — PASCAL
U.C4	Mini-monitor, disk resident, many useful functions — ASM
PRINT.C4	Parallel printer driver, without PFLAG — ASM
SET.C5	Set printer modes — ASM
SETBAS1.C5	Set printer modes — A-BASIC
	(And many more)

Over 30 **TEXT files included in ASM (assembler) — PASCAL — PIC (position independent code) TSC BASIC-C, etc.

NOTE: .C1, .C2, etc. = Chapter 1, Chapter 2, etc.

This will be a limited run and we cannot guarantee that supplies will last long. Order now for early delivery.

Foreign Orders Add \$4.50 S/H

Softcover — Large Format

Book only: **\$7.95** + \$2.50 S/H

With disk: 5" **\$20.90** + \$2.50 S/H

With disk: 8" **\$22.90** + \$2.50 S/H

See your local \$50 dealer/bookstore or order direct from:

Computer Publishing Inc.
5900 Cassandra Smith Rd.
Hixson, TN 37343
(615) 842-4601

™ FLEX is a trademark of Technical Systems Consultants

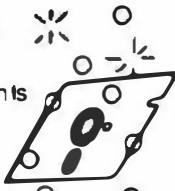


BUBBLE MEMORY

HIGH RELIABILITY STORAGE
FOR THE SS -50 BUS

DISK BUB provides 128K bubble memory replacing a disk drive.

- no moving parts to wear out
- direct boot capability
- withstands harsh environments such as dust, heat, vibration
- faster than standard drives
- utility software supplied



Diskbub plugs into the 30 pin I/O bus.
Flex09 drivers and boot rom provided.
Available for only

\$845⁰⁰



2457 Wehrle Drive, B-68
Buffalo, New York 14221
Phone (716) 631-3011

Dealer Inquiries Welcome

Business Software For The COLOR COMPUTER

WITH FLEX* & XBASIC

Data Base Manager

Part I _____ \$ 49.95

Part II* _____ \$ 49.95

Church Contributions _____ \$ 49.95

Balanced Billing System _____ \$ 49.95

Single Entry General Ledger _____ \$ 49.95

Integrated Business Software*

*available for Color Computer only

Accounts Receivable _____ \$ 99.95

Accounts Payable _____ \$ 99.95

General Ledger _____ \$189.00

Inventory 2 _____ \$ 69.00

Payroll _____ \$ 99.95

64K memory upgrade
including installation _____ \$ 79.00



Call or Write for free catalogue

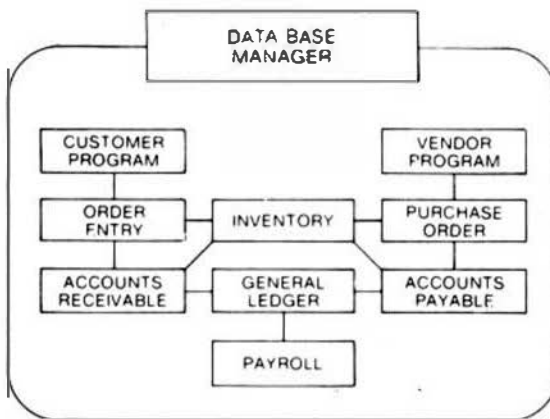
2457 Wehrle Dr., D-80, Buffalo, NY 14221
PHONE (716) 631-3011

Dealer Inquiries Welcome



417

FLEX* and UniFLEX* software for the 6809



Integrated Business Programs

	FLEX	UniFLEX
Accounts Payable _____	\$295	\$395
Accounts Receivable _____	\$295	\$395
General Ledger _____	\$295	\$395
Inventory 2 _____	\$295	\$395
Payroll _____	\$295	\$395
Data Base Manager _____	\$350	\$450
Word Processing		
Software _____	\$295	\$395
WP Menu _____		\$150
P Control _____		\$150



*FLEX & UniFLEX are Trademarks of Technical Systems

2457 Wehrle Drive, D-68, Buffalo, NY 14221
PHONE (716) 631-3011

Dealer Inquiries Welcome



TMP (Total Management Planning) OS-9 SOFTWARE

THESE OFF-THE-SHELF APPLICATION PACKAGES HAVE CHANGED THE PICTURE
FOR PROFESSIONAL SOFTWARE USERS

Here are the comprehensively supported OS-9 application software packages that discerning professionals are selecting today. Each one runs stand-alone or in combination with other TMP packages.

TMP/CALC is a new generation "Electronic Spreadsheet" package that goes a quantum leap beyond ordinary Calc software. Superior speed, and extra features like Dynamic Calculations and Dynamic Overlay, mean you can change a cell anywhere on the spreadsheet and get automatic updates wherever they belong. No more "wrap-around" since output to the printer is formatted. "Help" screens prompt without erasing data, and the rows and columns on any worksheet are limited only by available memory.
ONLY \$250

TMP/Manager is a structured database, manager, and is the ultimate driving force behind any total Management/Marketing Planning system. Its built-in select/sort/merge module — and optimized machine code — makes it super-fast, and provides unusually nimble data manipulation characteristics. Other outstanding features are detailed prompting, easy identification of data fields, and an excellent security system.
ONLY \$500

TMP FreeForm fills the void between a structured DBMS and word-processing. It's an "electronic index card" package with an endless list of valuable applications such as parts cataloging, look-up inventories and listings of any kind. Type anything on up to 32,000 electronic "cards" of up to 9 pages each that reside in "file drawers" (databases). Up to 13 key-words can call each page.
ONLY \$150

TMP/Front-End integrates any or all of the packages described above, and integrates TMP with word-processors, BASIC and other high-level languages. NO CHARGE with TMP/Manager



TMP PACKAGES ARE EXCEPTIONALLY FRIENDLY. EACH COMES WITH MULTI-MEDIA TRAINING AND OUTSTANDING DOCUMENTATION

AUDIO CASSETTE TRAINING TAPES are provided with each TMP software package. These crisp, professionally produced training aids take a new user

through "hands-on" exercises that instantly build confidence, and minimize reference to manuals.

CLEAR DOCUMENTATION is specially prepared for each TMP package. Remarkably readable because they are conversational in style and logically organized, this series of manuals further assures quick user comfort and productivity with TMP software. Includes sections on how to best utilize the system; integrating user-written software with TMP packages, and other bonus topics.

VIDEOTAPE PROGRAMS are available to TMP Dealers. They dramatically take the viewer into real business environments for a close look at how a TMP software package is utilized. Video programs can be personalized for specific organizations. Programs may be specified for most popular tape formats.

THE BOTTOM-LINE IS THIS: Software is only as effective as its training and documentation. TMP packages are incomparable in both areas.

RUN TMP IN A SMOKE SIGNAL CHIEFTAIN™ COMPUTER TO EXPERIENCE THE KIND OF SPEED AND POWER YOU CAN EASILY AFFORD TODAY



Endurance-Certified Chieftains are consistently among the fastest computers in CPU and I/O speeds according to the widely respected BENCHMARK REPORTS. That, combined with almost legendary reliability, makes Chieftains the logical choice in computers using OS-9 operating systems.

Configurations range from floppy-disk systems to multi-user, multi-tasking Winchester hard disk systems with tape back-up; performance approaching mainframes at prices you can live with.

TMP software packages for most OS-9 systems and Chieftain computers are offered exclusively from Smoke Signal. Inquiries from non-Smoke Signal dealers are invited. Call (213) 889-9340.

TMP T.M.



SMOKE SIGNAL
Chieftain Computers

31336 Via Colinas • Westlake Village, CA 91362
(213) 889-9340

► **ANOTHER WINNER! AAA Chicago Computer Center** is proud to announce the addition of the **HELI** line of computer products. The **HELI** line, together with **SWTPC**, **SSB**, and **ELEKTRA** permits us to offer you the greatest selection of quality components to handle all of your needs.

64K 6809 Computer	\$2395.00	256K 68000 Computer	\$3695.00
256K 6809 Computer	2895.00	512K 68000 Computer	4195.00
Other systems available			
20 Megabyte 5" add on Winchester system	\$2595.00		
64K CMOS Static memory board with battery backup	395.00		
DMA 5" and 8" Floppy Controller with built in Winchester controller I/O	695.00		
DMA 5" and 8" Floppy Controller	\$495.00	6809 CPU Board	495.00
68008 board for SS-50	495.00	68000 CPU Board	995.00

► Need FLEX, UniFLEX, OS-9 Level I, or OS-9 Level II? We have a system for you!

► **ELEKTRA COMPUTER CABINET** THE LARGEST SS-50 COMPUTER CABINET AVAILABLE! Made of heavy weight 0.090" thick aluminum. Interior is 18 1/2" wide by 21 7/8" deep by 6 3/4" high. Heavy duty A.C. line cord. A.C. fuse holder. EMI filter. Fan with filter. Back panel has 10 cutouts for D type data connectors. Front panel has key on/off power switch, 2 illuminated push button switches (Reset and NMI/Abort), and two cutouts for 5 1/4" disk drives. \$250.00

Filter Plate for 5 1/4" drive opening. \$10.00 Fan Filter. \$10.00

POWER SUPPLY Highest quality linear power supply CONSERVATIVELY rated at 15a @ 220v. 3a @ 16v. 3a @ 16v. 3 primary inputs for light, rated, and heavy loads. \$175.00 110v Version. \$200.00

DISK REGULATOR BOARD WITH CABLES Standard version for 2 floppy drives \$50.00 Heavy duty version for 1 Winchester drive and 1 floppy drive. \$75.00

► **ENGINEER'S "FUN BOX" BY ELEKTRA** Computer cabinet with high quality 10 amp power supply. EMI filter, fan. Large enough to hold the standard size SS-50 type motherboard. \$225.00

ELEKTRA UNIVERSAL SS-50/SS-50C MOTHERBOARD Heavyweight 0.125" thick. 18" long by 9" wide. 11 memory (50 pin) slots. 8 I/O (30 pin) slots. Complete address decoding and selection for I/O slots. Choice of 4, 8, or 16 addresses per I/O slot. 1" spacing between all memory and I/O slots. On board baud rate generator with low and high range jumper selectable rates of 75 through 38,400 for each of the five baud rate lines, slow device circuitry permitting 1 Mhz 30 pin disk controllers to run with 2MHz 50 pin CPU boards.

Mounting hardware. \$5.00 Bareboard w/documentation. \$80.00
Kit w/gold connectors. \$320.00 Assembled w/gold connectors. \$380.00
Kit w/in connectors. \$240.00 Assembled w/in connectors. \$300.00

ELEKTRA CHASSIS Includes cabinet, 110v power supply, power supply cables, standard disk regulator board with power cables, motherboard with gold supply pin connectors, assembled and tested. \$850.00

ELEKTRA CPU 8/9 Use either the 6802 or 6808 (to run 6800 software) or 6809. Has provision for up to 3 2716 EPROMs, 1K scratchpad, MC6840 triple timer, and an optional baud rate generator providing baud rates from 110 through 38,400 baud in two user selectable ranges. Versions of OS-9 level I are available.
Bareboard. \$50.00 Kit. \$225.00 Assembled. \$275.00
Optional Baud Rate Generator. \$25.00

ELEKTRA DPS DUAL PORT SERIAL CARD Fits the standard 30 pin SS-50 bus I/O slot. Can be configured for 4 or 16 addresses per port. RTS, CTS, DTR, DCD, IRQ, FIRQ, NMI, and baud rate can be appropriately implemented for each port.
Bareboard. \$20.00 Kit. \$60.00 Assembled. \$80.00
Cable with mounting hardware (two needed per board). Each. \$25.00

ELEKTRA DPP DUAL PORT PARALLEL CARD Fits the standard 30 pin SS-50 bus I/O slot. Can be configured for 4 or 16 addresses per I/O slot. The direction of the TTL buffers can be controlled by either on board jumper connectors or by a signal from the peripherals. The interrupt request line for each port may be individually jumpered to either the IRQ or FIRQ/NMI bus line.
Bareboard. \$20.00 Kit. \$60.00 Assembled. \$80.00
Cable with mounting hardware (two needed per board). Each. \$25.00

ELEKTRA 64K STATIC RAM/ROM MEMORY BOARDS with gold connectors (tin available). Assembled and tested. With 56K RAM \$299.00 With 64K RAM \$349.00

► **ELEKTRA SUPER FLOPPY CONTROLLER** THE BEST 30 PIN FLOPPY DISK CONTROLLER THAT YOU CAN BUY! Controls up to four 5 1/4" drives and four 8" drives for a total of eight system drives. (FLEX system limit is four drives.) Single density or double density, 1MHz or 2MHz, 6800 or 6809. (Double density 8" must be at 2MHz, all other combinations of performance are possible.) Analog phase locked loop data separator with separate adjustments for 5" and 8" drives. Designed to meet the data hold requirements of Western Digital floppy controller IC. Assembled and tested \$275.00
Disk with drivers and formatting utilities (Specify 6800/9, FLEX/OS-9). 30.00

► **ELEKTRA WINCHESTER SYSTEMS** THE BEST WINCHESTER SYSTEMS THAT YOU CAN BUY! Has automatic error detection and CORRECTION of up to 11 bit burst errors. SS-50 bus, extended addressing capabilities, DMA, on board sector buffer, drivers included for 6809 FLEX or OS-9. Specify whose version of FLEX that you are using. Drivers for FLEX2 (6800) are available for an additional \$100.00. Price includes host interface, controller, drivers, and cables.
12 Megabyte single drive sys. \$2995.00 24 Megabyte dual drive sys. \$3595.00
19 Megabyte single drive sys. \$2995.00 24 Megabyte dual drive sys. \$4695.00
(19 Megabyte drives are the largest that can be supported by FLEX)

ELEKTRA HD-5 Cabinet for dual 5 1/4" floppy drives with power supply, line cord, fuse power switch, and power cables to drives. \$150.00

ELEKTRA HD-5W As above but with EMI filter, fan, and heavy duty power supply. Powers 1 floppy and 1 Winchester. \$199.00

5" ribbon cable for dual 5 1/4" disk drives. 40.00

ELEKTRA HD-8 Dual drive cabinet. EMI filter, fan, power supply, and power supply cables for 8" drives. 350.00

6" ribbon cable for dual 8" disk drives. 45.00

ELEKTRA 30 PIN PROTOTYPING BOARD 20.00

ELEKTRA 50 PIN PROTOTYPING BOARD 40.00

GOLD 10 PIN CONNECTORS (Specify male with square pins or female) 1.50

TIN 10 PIN CONNECTORS (Specify male with square pins or female) 50

ELEKTRA is a trademark of AAA Chicago Computer Center.
FLEX and **UniFLEX** are trademarks of Technical Systems Consultants, Inc.
OS-9 and **BASIC09** are trademarks of Microware Systems Corp.
GIMIX is a registered trademark of GIMIX Inc.

Prices and inventory are subject to change without advance notice.
These ads are our catalog

AAA CHICAGO COMPUTER CENTER
120 CHESTNUT LANE • WHEELING, IL 60090 (312) 459-0450
Technical consultation available 4 PM to 6 PM most weekdays. Closed evenings and weekends.

TERMS Minimum order \$20.00. Shipping and handling estimates within the Continental U.S. add 3% (MINIMUM \$2.00). Illinois residents add 6% sales tax. We will refund you overestimated shipping and handling charges. Foreign shipping and handling, add 10% (MINI UM \$10.00). Foreign orders must be prepaid in U.S. dollars. Heavy foreign items will be shipped air freight collect. Please phone between 4 PM and 6 PM weekdays if questions arise regarding shipping fees. Master Charge, Visa, and American Express honored.

Our apology: We are not staffed to answer technical inquiries through the mail. Please phone for technical help during the hours indicated above. The too frequent changing of our inventory and prices makes it uneconomical to publish a catalog. Our ads are intended to serve that purpose. Prices and inventory are subject to change without advance notice.

ELEKTRA™ SOFTWARE (All of our software is copyrighted and all rights are reserved. Source is either supplied or optionally available at extra cost so that the purchaser can modify our programs for his own use. Licensing, however, is required for commercial resale.)

SUPER MODEM PROGRAM Single character commands. No interrupts required. Transmit manually or transmit disk files (text) of any length to distant computer. Receive and save disk files (text) on local disk system. X-on/X-off supported. Tested for full duplex at speeds up to 9600 baud. Half duplex option. Echo option. Replaces CR with CR/LF (user option). Slow disk file transmit option.

Please specify 6800 or 6809, SSB or FLEX™, 5" or 8"
Instruction Manual and disk with both source and object code \$75.00

OS-9 Modem Program 100.00

STANDARD MODEM PROGRAM Same as Super Modem Program above but without ECHO option, CR/LF for CR option, slow disk file transmit option, nor X-on/X-off option. Specify 6800 or 6809.

Manual with instructions, source listing, and flow chart 10.00

ORDER — WRITE UP COMPUTER PROGRAM Screen oriented write up form with cursor editing, disk save and load, printer command using easily available universal print-out forms. Phone for more details. Available for 6809 FLEX. \$100.00

ALL IN ONE

Editor — Text Processor — Mailing Labels — Mailing Lists
Use any CRT terminal and printer — Multiple Form Letters
Best Package For The Money Anywhere!

Specify 6800 or 6809, SSB or FLEX™, 5" or 8"
Printed source listing is available for an additional 35.00
All-In-One, Write'n spell, and Spell'n Fix package 250.00

Software by Technical Systems Consultants, Inc.

	UniFLEX™ w/1 yr. mnt	FLEX™
DOS (includes Editor and Assembler)	550.00	150.00
Editor or Assembler		50.00
6800 Cross Assembler on 6809	300.00	250.00
6809 Cross Assembler on 6800 or 6800 FLEX™ Utilities		100.00
Text Processor or Sort/Merge Package or 6809 FLEX™ Utilities	150.00	75.00
Extended Basic	200.00	100.00
Basic Precompiler (specify standard or extended)	150.00	50.00
Pascal	300.00	200.00
Debug Package or Diagnostic Package	75.00	75.00
6809 Relocating Assembler & Linking Loader	175.00	150.00
Fortran (Requires Relocating Assembler & Linking Loader)	350.00	275.00
Fortran (With Relocating Assembler & Linking Loader)	450.00	375.00
Cobol	750.00	500.00

Software by Microware Systems Corp.	Run-Time Package	Update	Source	Manual Only	Object w/Man.
OS-9™ Level One Operating System	75.00	400.00	N/A	40.00	200.00
OS-9™ Level Two Operating System	75.00	N/A	N/A	40.00	500.00
BASIC09™	100.00	75.00	N/A	25.00	200.00
OS-9™ Macro Text Editor		300.00	N/A	15.00	125.00
OS-9™ Interactive Assembler		300.00	N/A	10.00	125.00
OS-9™ Interactive Debugger (Disk version)		100.00	N/A	10.00	50.00
CIS Cobol Compiler	400.00	50.00	N/A	80.00	900.00
Pascal Compiler	100.00	100.00	N/A	40.00	400.00
C: Compiler		100.00	N/A	40.00	400.00
Microware yearly support service (\$200.00 for OS-9 Level 2)					
					75.00

Special Software

MICROBUG (2K, 6809 Baby HUMBUG by Peter Stark) 30.00
4K 6809 HUMBUG 75.00
4K 6800 HUMBUG (RAM needed at \$A000 and \$D000) 65.00
2K 6800 HUMBUG (With cassette LOAD and PUNCH) 40.00
2K 6800 HUMBUG (Extra commands instead of cassette software) 40.00

Other HUMBUG versions including video versions are available

Spell'n Fix by Peter Stark 178.58
Write'n Spell by Peter Stark 75.11
All-In-One, Spell'n Fix, and Write'n Spell package 250.00
Dynalite Disassembler 60.00
SUPER SLEUTH Disassembler System (\$101.00 for OS-9 version) 99.00

SO/DO DISK DRIVES	2 heads	1 head	2 heads
30 day guarantee	CDC	MPI	MPI
5 1/4" 40 tracks	300.00	225.00	275.00
5 1/4" 80 tracks	375.00	275.00	350.00
MPI or CDC Service Manual (Specify 40 or 80 tracks)			25.00
8" 77 tracks DS/DD	Qume DT-8 \$550.00	Remex (Special)	350.00

SPECIALS

• U.S. Robotics 300/1200 baud auto dial/auto answer modem 399.00/499.00
• U.S. Robotics 1200 baud direct connect auto answer modem 399.00
• Hazeltine 1420 CRT terminal 375.00
• SSB BFD Floppy Disk Controllers (Version 3) Run FLEX2 or SSB DOS 15.00
• SWTPC 4K Memory \$15.00, 8K Memory \$40.00, MP-C \$50.00
• High speed tape reader 50.00 SWTPC MP-A2 125.00
• 300 Baud acoustic modem 129.00 Ribbons for MX-80 5.99

GIMIX CLEARANCE SALE

	OUR LIST PRICE	OUR LIST PRICE
Relay driver board	538.86	250.00
Cable (Ser or Par I/O)	24.95	20.00
#58 floppy controller	226.58	175.00
Double disk reg. card	68.22	50.00
32K memory board	175.00	58K memory board
64K memory board	478.67	450.00
80 X 24 Video Boards	250.00	Dual prt ser. 1 cable
64 X 16 Video Boards	198.71	100.00
16K Mem Bds. most w/cntrl reg	145.00	Reset/Abort Toggle Switch
93L422 RAM chips (2 needed for GIMIX DAT)		each 20.00

SWTPC

6809 SWTPC FLEX™ and manual 35.00
DC-4 Disk Controller (SS/DS, SD/DD, 5 1/4") 230.00
MP-S2 Dual Port Serial 120.00
MP-N Calculator Board (kit) 54.95
MP-N (assembled) 92.00
MP-R 2716 EProm Programmer 114.50
MP-09 2MHz 6809 CPU Board 295.00

Smoke Signal Broadcasting

DCB-4A Double Density Controller Board for 5" and 8" with DOS 549.00
SSB DOS (Specify 6800 or 6809, BFD or DCB-4A, 5" or 8") 75.00
SE92/SA92-5 (6809 Edit/Assm for DOS) 69.95
SSB Monitor (Specify 6800/6809, \$8008/\$E008/\$F7E8) 75.00
SSB version of FLEX™ (without Editor and Assembler) 150.00
LMB-1A Motherboard 399.00
SCB-99 6809 CPU Board 399.00
PAR-1 Dual Port Parallel Board 89.00
SER-2 Dual Port Serial Board with 2 Cables 129.00
Cable 90 64K Computer System 2195.00

WARNING AAA Chicago Computer Center does not provide repair or diagnostic service for customer assembled kits. AAA Chicago Computer Center does warranty and maintain service for our assembled boards. The customer should carefully take into consideration the small differential separating out kit and assembled prices when making his choice of purchase.

We have introduced our line of computer equipment with the purpose of offering the highest quality of components possible at affordable prices. These products are intended for OEM applications where it is the responsibility of the purchaser to integrate these components with suitable memory, disk controllers, drives, and software along with I/O terminals to form working computer systems.

Color Micro Journal™

5900 Cassandra Smith Rd. ★

Hixson, TN. 37343

COLOR MICRO JOURNAL™ Is A Monthly
Tabloid Publication for Color Computer USERS!

★★

COLOR MICRO JOURNAL™ is a Magazine FOR
Color Computer Users BY Color Computer Users. Col-
umns on various compatible Operating Systems, Lan-
guages, Uses (Bulletin Boards, Clubs, using the RS
BASIC, and so on), etc.

★★★

Programs - Games - Reviews - Education - Hardware -
Software - New Product Announcements - Books

★★★★

Get the MOST from your COLOR COMPUTER
without being an Engineer.

DON'T MISS A SINGLE ISSUE

Subscription Rate of only \$16.50 a Year!!!

★★★★★

COLOR MICRO JOURNAL™ published by the
ONLY pure 68xx, INTERNATIONAL Computer Maga-
zine. '68 Micro Journal has provided coverage for over
FIVE Years. We KNOW the Color Computer, the Soft-
ware (both FUN and WORK) that IS and CAN BE run on
it. We KNOW the products that ARE, CAN BE, or
WILL BE used on the Color Computer.



Color Micro Journal

Limited Time Charter Rates

USA - \$16.50 per year. Canada & Mexico - \$23.00 per year

Surface Foreign - \$28.00 per year. Airmail Foreign - \$52.00 per year

*Color Micro Journal is a trademark of Computer Publishing Inc.

**For Ordering Subscriptions By Phone
Call 1-800-338-6800**

.....
• Yes! Start my copy of Color Micro Journal coming as soon as possible!

• Name _____

• Address _____

• City _____ State _____ Zip _____

• ☐ Visa ☐ Master Card ☐ Check or Money Order Enclosed

• Card # _____

• Exp/Date _____

Flex User Notes

Ronald W. Anderson
3540 Sturbridge Court
Ann Arbor, MI 48105

MY TURN

A few months ago I made some remarks about someone else using my published scientific functions without any acknowledgement of my authorship. Now it is my turn to apologize. In my book "From BASIC to Pascal" I used some of Lucidata's demonstration programs (with permission from Nigel Bennet of Lucidata). Seems that I mentioned Nigel's name in the book but didn't mention Dave Gibby, who, it turns out, wrote the demo programs. I'll have to plead ignorance of authorship since Dave's name does not appear in any of the demo programs, and there was no indication anywhere in the documentation that he had written them. I was aware that Dave wrote the Runtime Interpreter for the Lucidata Pascal, and should have mentioned his name on that basis, however.

Dave, I can only say I'm sorry. I didn't dwell a great deal on Lucidata in the book since I was trying to write it generally to apply to any of the microcomputer Pascal implementations including those for other processors. I might add that though I mentioned OmegaSoft, I didn't mention Bob Reimiller, Al Jost, Dynasoft, or TSC by name either.

SECOND GENERATION

As most of my readers know, I put together hardware and software for machine controls and measuring instruments for a living. (I generally preface a discussion with this for the benefit of new readers). An idea has slowly come to us at our company, that we can make an instrument smarter. Let me take a simple example and explain what I mean. Suppose you were a builder of scales (weighing devices) and that some years ago, you had modernized your line of scales by using a transducer and a digital voltmeter. Of course, you had to provide some sort of "drive" for the transducer, and most likely amplify its output and process it in some way for the digital voltmeter.

Now of course, the drive to the transducer must be very stable, and you must carefully adjust the signal gain of your signal amplifier to get the scale to read exactly 1.000 with a weight of 1 Kg. In place on its weighing platform. You must also adjust the "offset" so that the scale reads 0.000 with no weight on the pan. What we've done, though in an area more complex than a scale, is to make the system self calibrating. In terms of the scale, we tell it that there is no weight on it and it should read zero. Then we put a precision calibration weight on it and tell it what that weight is. The processor then reads the actual voltage it is receiving for zero weight, and for the calibration weight, and it calculates the offset and the volts per gram constant for itself.

If the instrument has some battery backed up RAM memory, it can store these constants until someone wants to check its calibration or recalibrate it, at which time it will measure the voltages and calculate the necessary constants that it needs to display the correct weight.

It doesn't matter a great deal whether the transducer produces an output that is linear with weight, proportional to i/weight , weight squared, or whatever. As long as we know the relationship, that is, we have a good mathematical model of the transducer, we can solve for the necessary constants.

Perhaps it might be necessary to take three readings, no weight, 500 Gm. and 1000 Gm. in order to calculate the constants for a non-linear system, but that only adds one extra step to the procedure. If we were to have to input the constants via "trimmer potentiometers", we would likely find out that one adjustment interacts with the others, and we would have to go through the zero, half scale, full scale adjustments several times in order to get the system tuned to track the transducer. If we can describe the transducer response with an equation that contains two or three unknowns, we can take some readings with known stimuli and calculate the values for those unknowns, and henceforth have an accurate measurement of whatever we have set out to measure. We have applied this approach to one instrument, and it has performed beyond our greatest expectations.

HOW OLD IS OLD?

This might turn into a long story, but the question for today is whether FLEX should be or has been kept "standard" over a long period. I received a letter today from a reader, (or rather a copy of a letter to Windrush Micro Systems) describing a peculiar problem the reader had in getting PL/9 running on his 6809 system. It seems that way back in FLEX2 times, FLEX had an "echo flag". This byte determined whether FLEX would echo back to the terminal or not. When FLEX9 came along, this flag had been eliminated. It seems that the newer versions of FLEX9 have in addition to the standard drivers for terminal input with echo and output, a third "input character no echo" routine.

The problem was that there were no more jump vectors left in the area of \$CD00 where all the standard FLEX routine jumps have always been. PUTCHAR for example is entered at \$CD18. GETCHAR is entered at \$CD15. These addresses are invariant for all versions of FLEX9. The reason for their existence, in fact, is to provide a "constant" entry point for the programmer usable FLEX routines.

It might be interesting to trace through GETCHAR just a bit and see what happens. The address \$CD15 simply jumps to another location inside of the body of FLEX. Version 2.7:3, which I happen to have, has a jump to \$CEFB. The code there does a variety of things. First of all, input could possibly be from a file rather than the terminal. The routine tests to see if the byte at \$CC23 called Input Switch is 0 or non zero. If it is zero, GETCHAR jumps to the vector INCH at \$CD09. INCH may be modified by the user program so that it can be a jump to a routine to input from a file, etc. If SWITCH is non-zero, GETCHAR jumps to INCH2 at \$CD0C. This jump always points to the terminal input routine. Any program that uses a file for input (for example, the L.CMD utility switches input to a file, as does the EXEC.CMD) can either call the FLEX routine RSTRIO, or reenter FLEX with JMP WARMS which will automatically set the jump for INCH to be the same as that at INCH2.

PUTCHAR works the same way. Output is normally passed by it to OUTCH at \$CD0F, but if Output Switch is on, output goes to OUTCH2 which always points to the terminal output routine. OUTCH might be changed by calling for output to a printer via the P command, for example. When PRINT.SYS is loaded, it overlays the jump vector at OUTCH to point to the printer output routine. Again exiting the program via a jump to FLEX WARMS will restore the output jump to the terminal output routine.

For some time now, FLEX has been available in a version that allows the user to write his own I/O drivers so that it may be used with a different monitor or hardware configuration. General FLEX instructions document some other jump vectors in FLEX9 that are user changeable that will allow the user to point the FLEX I/O to his own monitor. In Standard FLEX, (mine is version *2.7:3) INCH normally does a jump to an input routine at \$D383 which simply does a jump indirect to the address found at address \$F806. Now this seems rather involved for getting to the input character routine. \$F806 is an address that is within the SBUG-E ROM, and it is called INCH in the SBUG manual. This indirect jump gets to the actual code that inputs a character from the terminal. SBUG-E expects the terminal to be connected to a serial port at address \$E004.

OUTCH jumps to a routine at \$D37F which does an indirect jump through SBUG at \$F80A which SBUG calls OUTCH. What I am leading up to is that somewhere along the line, a number of "screen oriented" editors came along. It became necessary to be able to input a character without echoing it back to the terminal. TSC added a jump vector at \$D3E5 which points at \$F804 in SBUG, which is INCH without echo. My very oldest version of FLEX, called FLEX 9.0, my original disk from TSC dated 10/10/79, does not have this jump in it. In fact the date at \$D3E5 is \$E004, the address of the port itself. I checked FLEX9 version 2.7:3, which was not much later, and it does have the proper jump at \$D3E5 to get to the INCH without echo.

The reader had written to Windrush that PL/9 had not worked, and that he had found a number of references to \$D3E5 which contained the hex value E004. This reader was able to find the references in PL/9 to \$D3E5, and patch them to his own input routine without echo. The letter was simply a suggestion to Windrush that they document the expected I/O routines so that someone with a "non standard" system might be able to adapt PL/9 to it. Version 2.7:3 of FLEX9 has the user vectors spelled out in the section on General Flex, but INCH no

ecno is not included at least in the copy of the manual that I have. It is, as I said, however, built in to the standard 2.7:3 version. It would appear that this addition was made about three years ago. The problem I mention here was obviously a "no fault" situation. The software should work with FLEX on a 6809 system. The user had FLEX and had been running it for a long time on his system. The supplier didn't research thoroughly as to how many versions back FLEX would work correctly with their product. I don't think that should be expected, though it would be appreciated if the manual would say "This software requires FLEX9 version X.X:Z or newer to operate properly."

New software suppliers are beginning to sell their wares every month. Some of them hadn't thought of going into business in 1979 when that first version of FLEX9 was released. Should we expect them to test their wares with three or four year old versions of FLEX? I would say no to that. Though it would seem to be an expensive proposition, I guess we computerists can't expect the world to remain stationary at any one point. It is not unreasonable to have to purchase a later version of our favorite operating system after three or four years, or to expect to have to patch something once in a while to accommodate software written for newer versions. The main problem is figuring out what has changed so that the problem can be taken care of without expenditure of a great deal of time and energy.

Dynastar

I recently had a chance to try out Dynastar, the editor from Dynasoft Ltd., by Al Jost. Unfortunately the companion Dynaform text formatter is presently only available in OS9 form. However, Dynasoft by itself can prepare a formatted text file that is properly filled and justified. It can center titles, indent paragraphs, "outdent" titles (Dan Farnsworth, pardon my coined word again), and in general do everything but paging and page titling. It turned out that I could run the text through my homebrew text formatter **JUST and get pagination and page numbers. JUST only requires one command to be embedded in the text file as prepared by Dynastar, and that is the signal to my Epson printer to turn on the Emphasized print model. Actually that could be done manually so the file could for all practical purposes be run through JUST with no commands within the text. The display on the screen, except for page breaks, would look exactly like the final printed material.

When I received Dynastar, I was of course in a hurry to try it out. I found that I had to prepare a short Assembler file called GOTOXY to match Dynastar with my terminal. Simple but not idiotproof.. After a couple of tries, I read the manual. "Edit one of the GOTOXY files supplied on the release disk if your terminal is not represented" I have an old ADM-3 and it was not included. I found the Televideo 925 file to be very close, and I edited and assembled it. "The file must be on the system disk and be renamed GOTOXY.SYS." I did that but no luck, the screen filled with garbage. Nothing else to do but read the instructions to completion. "The GOTOXY.SYS file must have a special format. You must GET your assembled binary file **GET GOTOXY.BIN and then save it with the utility supplied **SAVESYS,GOTOXY.SYS,0,0100,0132. Gee, funny thing. I did that and Dynastar suddenly started working perfectly.

The above only proves that I too get excited when some new software arrives. No, I haven't learned to read the instructions thoroughly FIRST. I have learned to copy the software from the supplied disk to a "test disk" for experimentation, however, and I do that without fail.

What is Dynastar like? Well, I'm not going to give you a blow by blow description of all the control codes. I will say that all are present when needed in a list that takes up about four lines at the top of the screen. When you have memorized all of them, which should take a few weeks of daily use, you can "foggle" them off the screen so they no longer appear.

The good news about this editor is that it is very much like Wordstar. Good news because it will be easy to learn for any of you who work with CP/M systems at work and have learned wordstar. Dynastar is not quite the whole Wordstar, but it is a totally adequate subset. It allows insertion of text from a file, writing portions of text to a file, writing a file and loading another one while staying "in Dynastar". You can access FLEX to do a directory or delete a file, and you can edit a file longer than the edit buffer. You can define a macro, i.e. a series of characters to be sent to the editor when a

key of your choice is pressed. These are handy for defining such operations as global changes, to change all occurrences of "Mnemonic" to "Mnemonic" when you find that you had spelled it wrong in a 10,000 word paper!

The cursor is controlled by a convenient cluster of keys that are operated with the left hand, grouped close enough to the "control" key so that you can operate that with your little finger while finding the appropriate control key with another finger. You can move the cursor, of course in four directions by one character. You can move it left and right by one word. You can go to the top or bottom of the currently displayed screen, or to the left or right end of the current line, and you can move to the top or bottom of the whole edit buffer.

You can also move the cursor to the nearest "TAB STOP" which appears on a "ruler line" at the top of the text. You can set the width of the text, and set tab stops individually. If you have JUSTIFY or WRAP mode on, you simply type along with no regard to the end of the line. Dynastar moves a word that doesn't fit on a line, to the start of the next line. If you have JUSTIFY on, it immediately formats a line when it is finished. Should you make changes in a paragraph that is already formatted, you will of course change the length of some lines. No matter, just as in Wordstar, you put the cursor at the start of the paragraph and type CKA and the paragraph is reformatted magically. Should you decide to change the width of the whole file, you can define a macro to repeat the format command at the beginning of each paragraph until it runs out of paragraphs.

Dynasoft has an automatic indent mode, in which you may type in programs in your favorite high level language, be it Pascal, "C", PL/ or whatever. The cursor aligns itself with the first character of the previous line. To indent to the next tab, you simply type C. To back out to the previous one, you must backspace, which I consider a minor inconvenience. (Why isn't there a "backtab" key as well?)

This editor was written by Al Jost, as I indicated above. It is written in Dynasoft Pascal. You might be saying "That has to be too slow to be useful". Well, you are absolutely wrong. Al has incorporated some magic in Dynastar. It doesn't miss a character no matter how hard I try to type at absolute top speed at the end of a line when it is updating the screen. Al must have put some code in to make it look for and accept input characters into a buffer between writing characters to the screen. The text does get behind the typing, but it does catch up. The longest I've been able to stay ahead of the screen is about 15 characters at the start of a line. I type fast enough so that I can't use print spooling while I am dumping text in with an editor, and some of the screen editors are useless because they ignore input while updating. Not so with this one.

Oh, yes, I forgot to mention the variety of delete commands. You can delete the current character (cursor stays where it is and text to right moves to the left), delete the previous character (cursor moves left and character disappears), delete the WORD to the left or to the right (different commands), delete everything to the right of the cursor on the current line, or delete the entire line. Of course you can "mark" a block of text and delete the whole block. You can FIND a string or REPLACE a string with another. Of course you can copy or move a block of text.

Then we have the formatting commands, some of which we have talked about, such as JUSTIFY and WRAP. You can center the text on a line, set margins and set or clear tabs. You can also signal your printer to print BOLDFACE, Double strike, or to Underline text.

I think that about covers what you can do with Dynastar. I am not totally familiar with Wordstar, but I do know that there are a couple of features that are not here. For example, you can include a ruler line within a text file and have it not print. That is useful for setting up a table, for example, or a section of text that is narrow for the inclusion of a photograph, etc.

Assuming that Dynaform will be available in FLEX version, I should mention that you can embed commands for Dynaform in the text file. The friendly prompt list allows you to look at all the Dynaform prompts while editing. As I said at the start of this, I am not a "Wordstar whiz". I consider myself rather quick with PIE or Stylograph, but not with any other editor. I was able to learn to use Dynastar effectively in an hour or so of playing with writing a text file, a program, and reformatting some old text files that I had on a disk.

I generally hesitate to try new editors. An editor is a bit like a car. (If I smoked a pipe, I suppose I would use that analogy). We own two cars, and I can drive either without thinking too much about it. I suppose I could get used to driving a large car with automatic transmission and a small car with stick shift without too much trouble. However, increase the number beyond two or three, and you begin to have problems. I love PIE (see story below) for programming, and in a limited way for text editing (it doesn't format the text on the screen). I think Stylo is super for text when I want to see exactly what it will look like when I am done, but Stylograph has no "auto indent" feature.

Had I at this point to choose just ONE editor I would unhesitatingly pick Dynastar! I presently need the HELP prompts to keep all the commands straight. Generally, I find that 20% of an editor's capabilities are used about 96% of the time, and the others must be looked up to be used. After three or four years of using PIE, I still have to look up the procedure for writing lines 17 through 36 to a "side file" etc. Having those prompts to rely on, will make mastery of Dynastar much easier.

Dynastar is sold by Frank Hogg Laboratories.

The PIE Story

I mention PIE quite frequently, and it is time that I explain. PIE originally stood for "Programma Interactive Editor". It was written for the 6800 by Tom Crosley. Later, PIE was translated to 6502 code to run on the Apple(!) Still later, Tom translated it to 6809 code. About that time, several copies were distributed for evaluation, and I was lucky enough to get one. I have been told that a couple years ago, Programma sold out to Hayden Books. Their financial condition was so bad that Hayden decided to declare that division bankrupt. That effectively blocked any chance of PIE being marketed. I have lost track of the story from there, having heard that some nice manuals were printed but that PIE is not being distributed by anyone. If anyone knows more of the story, I would appreciate hearing about the current status.

Dyna-C

Dyna-C is "another 'C' compiler". It too, as the name implies comes from Dynasoft, and was written by Al Jost. Dyna-C produces Assembler source code from standard "C" source code. It is assembled by any of the "standard" 6809 assemblers, and produces fast object code. I ran the famous benchmark test with it, and the result was a 10 second time, as fast as the fastest "C" I have tested. Al coded a four line "outdec" procedure in his version of the benchmark, that allows not including one of the larger library files for formatted output. The result is that the benchmark object file is about 1250 bytes long, compared to about 6600 bytes for some of the other "C" compilers.

This is a "tiny C" implementation. Let me quote Al Jost from the manual. "Like many other low cost C implementations, Dyna-C owes its roots to the Small-C compiler published by Ron Cain in Dr. Dobbs Journal in June of 1980." Most of the other implementations have Structures and Unions. This implementation does not. Many of the others have multi dimensional arrays. This one has single dimensional arrays only. Some of the others have implemented long integers and some of the more advanced features such as "macro pre-processing" and unsigned integer. Having implied that the implementation is rather incomplete, I need to quote another section of the manual.

"Dyna-C is a pure subset of a full C, and any valid program acceptable to the Dyna-C compiler should be acceptable to a standard C compiler. The major omissions in Dyna-C are float, double, long, unsigned, structures, unions, bit fields, initializers, and macro arguments." I hasten to add that the last two of those items seem to be the very last to be implemented. I know of no C compiler that has bit fields implemented, and only one that has initializers. Within its limitations, as Al Jost points out, this is a completely standard C implementation. A great deal more of the standard library is written in Assembler than in many of the other implementations, and that fact makes this C faster and more compact.

Compiler options allow compilation of the program in parts so that pre-compiled assembler source code may be included for assembly. There is an option to leave the C source code as comments in the Assembler source file, which might be a great help in understanding the compiler or debugging a program.

I don't have to tell my regular readers that I am impressed with any compiler that generates efficient object code. This one definitely falls into that category. If you can get along with what is implemented, this is a good one. It too is available from Frank Hogg Laboratory.

* FLEX™ versions in this range are normally SWTPC versions. They normally encompass enchantments coded by SWTPC programmers, with utilities not available from TSC or their standard 6809 FLEX package.

** JUST - a formatter - is available from South East Media. See their advertising this issue.

COLOR User Notes

Robert L. Noy

5900 Cassandra Smith Rd.

Hixson, Tn. 37343

You should have received a copy of the new Color Micro Journal a few weeks ago; we will let that, and especially the THIS 'N THAT editorial, be this month's Color Users Notes. You should find the comments about OS-9 and the DRAGON DATA info interesting. Till next month;

--- RLN ---

OS9 USER NOTES

By: Peter Dibble

517 Goler House

Rochester, NY 14620

OS-9 by itself does very little useful work. You won't find an editor, assembler, compiler, spelling checker, or payroll system anywhere on the standard distribution disk. That isn't to say that you can't get these programs for OS-9, or even that some of them aren't sometimes packaged with the operating system (Gimix packages Microware's editor, assembler, debugger, Basic09, and RunB with every OS-9 system), but OS-9 can be purchased with no frills, and in that form it is essentially useless.

For an experienced microcomputer user with lots of friends using OS-9 and a nearby store with a large stock of OS-9 software the task of choosing the right array of software could be fun, but for me it was frightening. The least expensive software I could find cost about fifty dollars a crack, and it went up fast from there. I didn't know anyone running OS-9, and, though there were many computer stores in Rochester, the only one which dealt in 6809 based machines believed strongly (nearly exclusively) in TSC software. I gritted my teeth and bought what looked good to me. I was surprised to find that everything I bought was at least OK. In retrospect I can see that it wasn't so very surprising that I was lucky in my software purchases; most of the software for OS-9 is good.

With OS-9 I got the Microware Editor, Assembler, Debugger, and Pascal. I have no special love for the Microware Debugger, but I still use it because it is the only game in town. It usually is packaged with OS-9, and it is hard to get along without, especially if you do assembly language programming, but I hope Microware feels a touch of humiliation each time they send out a copy of that program — it is not up to the standard set by their other programs. The assembler is unexciting, but it does the job. There are other assemblers around, but the Microware assembler is the standard.

The Microware Editor is hard to classify. It is the only non-screen-oriented editor for OS-9 that I know of. It works fine as a simple editor, but it might be more accurate to call it a simple string processing language. The editor features multiple work spaces, and a high powered macro language which can be used to write fairly sophisticated programs. The bad side of all this sophistication is that it is a little bit hard to use the editor for simple things. I have never been able to figure out how to copy a range of lines without using a disk file as a temporary holding piece. I don't use the Microware Editor very frequently since I got a screen-oriented editor, but I got a lot of work done on it when it was the only editor I had, and I still use it occasionally. I should add that some people think editors like the Microware editor are better for programming than the more word processing oriented editors.

It is hard for me to be moderate in my praise for Microware's Pascal. I wish it included a debugger, and the procedure for linking to external procedures is a bit clumsy, but I love it. I use it to develop programs for classes where the students use DEC Pascal and IBM Pascal and have no compatibility problems. There are enough enhancements to make this Pascal useful for real applications (such as a PROMPT built-in procedure which forces out the contents of an output buffer without a carriage return). The compiler generates intermediate code which can be executed by either of two interpreters (one normal, and the other supporting large programs by a paging arrangement), or translated into efficient native code.

Recently I got Basic09. You may have guessed from my comments that I am getting to like it even though it is called Basic.

I have DynaStar, DynaForm, and DynaSpell from Frank Hogg Labs. None of these programs are exceptional, but I use them all regularly. DynaStar is a screen-oriented editor with which I have typed and revised many hundreds of pages. It is best at editing documents, but usable for programs. I expect the reason the program is called DynaStar is that it borrows heavily from Wordstar. My mother uses Wordstar, and I find that I can help her untangle some problems with Wordstar by assuming that it is keystroke for keystroke identical with DynaStar. I have some small complaints about DynaStar, but the bottom line is that I like it well enough to have spent hundreds of hours using it.

DynaForm is a text formatting/mail merge program. It is full of fancy Mail-Merge features that I never use. I use it to print files with optional page headers and trailers, underlining, and bold printing. A few times I have used its ability to generate indexes and a table of contents. DynaForm doesn't do well when compared to the high powered text formatting packages used on large computers, but I don't think it is intended to compete with that kind of thing. The thing about DynaForm that annoys me most frequently is that it can't be customized to use the special features of my printer. It prints bold text by simply printing the bold characters three times. DynaStar can be used to embed printer control characters in text, but DynaForm only knows one way to print bold or underlined text. I also wish it would use the standard input and output paths instead of allocating special paths.

DynaStar and DynaForm were written by Allan Jost. They show signs of being written by a programmer with a very professional attitude. They are not loaded with features but they are so reliable that I just take them for granted.

DynaSpell is a spelling checker. I need a spelling checker very badly. Some people buy computers to run a spreadsheet program. I might have bought one to run a spelling checker. DynaSpell essentially looks up each word in a document in a set of dictionaries. Any words that it doesn't find are treated as questionable words. These words can be fixed, accepted as is, or accepted and added to a dictionary. DynaSpell isn't as carefully written as the programs by Allan Jost; there is nothing major wrong, but the meticulous care isn't there. When DynaSpell runs out of space to store words in, it spews out pages of "overflow" messages. There is no way to check the contents of the directory when DynaSpell is asking for the name of the file to check. When you abort the program (with a control C) in order to check the directory again, DynaSpell leaves the terminal's device descriptor in a strange state. DynaSpell has most of the features commonly found in spelling checkers for microcomputers, but it doesn't compare with similar programs on larger machines. Maybe a spelling checker is one of those tasks which needs fast machines with large memories. I want a spelling checker which helps me correct misspelled words by giving me a list of suggested spellings, and a built in thesaurus would be another nice touch. Still, I use DynaSpell when it is inconvenient to ship my files off the the IBM to be checked. It isn't a great program, but it does its job.

I reviewed DynaCalc a few months ago. I still like the program, and it is still heavily used. I wouldn't have chosen DynaCalc as part of my core group of software (I mostly program and write with my computer) but I can imagine people who might not need any other program.

New Release of Microware Pascal

I just got release 2.0 of Microware's Pascal. It is a major revision, including a new intermediate code language, a single general purpose l-code-to-native-code translator, and new run time support modules. I didn't do any careful comparisons of the two versions, but I get the strong feeling that the new release compiles faster, and runs faster. The new manual is significantly better organized and more complete than the old one, but still makes no attempt to teach Pascal. Two new standard functions have been added: GETCHAR, which returns a single character from input, and IOREADY, which returns true if there is input ready. These new functions should be useful for interactive applications like editors.

OS-9 Directories

A directory is a special type of file containing information about files. It could be seen as something like a library's card file. It contains the names of files along with information about them, especially where they can be found. Unlike anything a proper library would contain, the entries in a directory aren't kept in any particularly useful order. You can get a formatted listing of the contents of a directory with the DIR command.

OS-9, like UNIX and many other multi-user operating systems, supports hierarchies of directories on disk. Directories can be used for a number of things, or, if you like, largely ignored. A directory can contain any number of other directories in addition to normal files.

Every disk has a root directory on it which is created when the disk is formatted, and cannot be done away with. Unless you fuss around with INIT

and SYSGO the disk you boot off of must have a directory called CMDS in its root directory. There may also be a SYS, and a DEFS directory in the root directory on the boot disk when you install OS-9.

You (the user) can create new directories with the MAKDIR command. To use the command type MAKDIR followed by the name of the new directory you want to create:

MAKDIR /D1/SOURCE.DIRECTORY It has become a convention to use capital letters for directories' names. OS-9 doesn't have any trouble with lower case directory names, but it is an easy way of reminding oneself which files are directories.

It is sometimes tricky to keep track of a library of several hundred (maybe thousand) files. Multiple directories are a major help in organizing files in such a way as to maximize the chance of finding them again. Long ago I found that I couldn't fit all my files on one disk (that was a 100K floppy back then). I put each major project on a separate disk. When I got disk drives with greater data capacity, I found that it wasn't an unadulterated good thing. Each disk contained so many files that it was a major job to locate a file even knowing which disk it was on. I worked out naming conventions that made the job easier, but they used up the first two characters of each file name -- the resulting file names were pretty cryptic. I still keep hundreds of files on each disk, but my largest directory has about forty files in it.

The root directory on a disk I have labeled "pascal" contains nothing but seven directories: DIST.SRC, UTIL.SRC, SUBR.SRC, BUGS, DEFS, DOC, and PCODE. Each of those directory names describes what I expect to find in them pretty well (to me anyhow). Each directory with programs in it contains a directory called DOC which contains related documentation. If it seems like I have large numbers of directories called DOC, it's true. Pretty near everything needs documentation. Sometimes I find that a directory begins to get out of control. Projects that I expect to need about ten files have a way of expanding to forty or fifty files. A project like that really belongs in a directory of its own, so I create a new directory in the directory that contains the files for the project, and move all the files that are part of that project into the new directory.

Any file can be accessed by giving its full name, e.g., /D1/UTIL.SRC/DFIX/Compacter would denote the file Compacter in the directory DFIX which is in the directory UTIL.SRC in the root directory on disk D1, but that's more typing than I would choose to do except as an act of desperation. The most commonly used shortcuts are the CHD, and CHX commands. The CHD command changes the directory which is treated as the root directory for data. CHX does the same thing for the execution directory.

When OS-9 is booted the data directory is set to the root directory of the boot disk, and the execution directory is set to CMDS in the root directory on the boot disk. If you want to use files in the root directory on the boot disk, all you need to do is give the file name, if you want to use files in a directory which is in that directory you give the name of the directory with the file name, e.g., to get at the file OS9Defs in DEFS in the data directory use DEFS/OS9Defs. If the default data directory isn't convenient for you, a new directory can be selected with the CHD command, for example, to change the data directory to the root directory on /D1 use CHD /D1. The CHX command works the same way CHD does, but it effects the execution directory.

There are two special entries in every directory. The "." entry points to the directory itself, and the ".." entry points to the directory the current directory is in, the parent directory. A typical use of the ".." entry is to refer to sibling directories. When a project gets large, I break it up into a set of directories, all in a directory which I set aside for the project. If a program needs access to the file HexDefs in the directory DEFS which is a sibling of the directory SRC (where the program is), I can use the shorthand name "../DEFS/HexDefs" for the file. I have found this a good convention to stay with. As long as I continue to keep related families of files in directories that are siblings, the notation "../DEFS" will always get me to the appropriate DEFS directory, and "../DOC" will always refer to the related Documentation directory.

To experiment with directories, start with a disk with some empty space on it, and use CHD to set the data directory to the root directory. Build some directories:

```
MAKDIR TESTD1
MAKDIR TESTD2
MAKDIR TESTD3
```

Make things a little more complicated:

```
CHD TESTD2
MAKDIR TESTD21
MAKDIR TESTD22
MAKDIR TESTD23
CHD TESTD21
MAKDIR TESTD211
MAKDIR TESTD212
MAKDIR TESTD213
CHD ../TESTD22
MAKDIR TESTD221
MAKDIR TESTD222
CHD ../TESTD23
MAKDIR TESTD231
MAKDIR TESTD232
CHD ..
```

CHD .. Now we're back at the root directory. The DIR command should show the files that were in the directory before you started this experiment plus the directories TESTD1, TESTD2, and TESTD3. DIR TESTD1 will show an empty directory. DIR TESTD2 will show the directories TESTD21, TESTD22, and TESTD23. The following commands will all show the contents of the directory TESTD23:

```
DIR TESTD2/TESTD23
DIR ../TESTD2/TESTD23
CHD TESTD2 ; DIR TESTD23
CHD TESTD2/TESTD23; DIR
```

The first two command lines leave the data directory at the root directory. The third command line moves the data directory to TESTD2, and the fourth command line moves the data directory all the way out to TESTD23.

It is easy to create new directories, but a little involved to delete a directory. Perhaps it is a good thing that it requires more than one quick operation to remove a directory. If a directory with files in it is erased, all the files in the removed directory will remain on the disk, but OS-9 won't be able to locate them. Older versions of OS-9 don't have any command which will delete a directory. To do away with a directory with these older versions: delete all the files (and directories) in the directory, use the ATTR command to change the directory into a normal file (ATTR <dirname> -d), and delete the file that used to be the directory. Be particularly careful not to use ATTR to change the directory into a regular file until the directory is empty. There is no easy way to change the file back into a directory so you can delete the files in it. With the new release of OS-9, the command DELDIR can be used to delete directories. DELDIR simply automates the steps I just went through.

Directories are an important feature of UNIX-like operating systems. They allow files to be grouped in manageable clusters, and make it easier to handle many concurrent users.

I am preparing to eat some of the words I set down in my first column. I am looking forward to this with a good deal of pleasure -- they were critical words. Some people have gone to a fair amount of effort to convince me that I was wrong. If things go well I'll hold the word eating ceremony next month.

STASM

Sansaska STASM Review

You say you like Pascal, 'C', PL/C because these languages allow you to structure your program? Maybe you have no idea what "structured programming" is all about and you'd like to find out. Maybe you'd just like to make your Assembler programs more readable. If you fall into any of those three categories, read on.

Sansaska's STASM is a way to make your assembler programs more structured. For those of you who might not quite understand what we're talking about, structured programming essentially is the use of some few basic "structures" rather than endless GOTO statements, to make a program more readable. I say few structures because there are really not very many. The first of these is perhaps the IF-THEN-ELSE structure. It looks like this:

```
IF <condition>
  THEN <do this>
  ELSE <do this>
```

That is, this structure is a convenient way to perform alternate actions based on a test or comparison of two parameters.

Another structure is the FOR-NEXT loop that is used in virtually all of the high level languages. It is a way to perform an action a predetermined number of times. I can't imagine any reader of '68' not being familiar with the use of this structure in BASIC, so I won't elaborate further.

Pascal, C, and Fortran have what is called a DO loop. Pascal uses the syntax:

```
WHILE <condition> DO <action>;
```

Pascal has another similar construct called the REPEAT UNTIL which looks like this:

```
REPEAT <action> UNTIL <condition>;
```

The difference is slight. The while loop may not be executed at all, since the test is made at the beginning, and if it is false, the action will be skipped. The Repeat loop, on the other hand will execute the action at least once, since the test is at the end of the loop.

The C language has a way of terminating a loop "prematurely" that is before the count or the condition is met (for example on an error) by means of something called a BREAK statement. Normally one would use a condition such as IF EOF THEN BREAK. That would test for an end of file condition. Break causes the program to jump to the statement after the end of the loop.

The point of all this is that STASM allows an assembler programmer to use all these structures and in fact allows a bit more flexibility than the high level equivalent of some of them. STASM is a couple of things. First it is a text pre-processor that modifies the listing format. Secondly it is a file of macros that supply the code for the high level structures.

I say that it changes the listing format. High level languages that use the above structures usually allow the programmer to indent in such a way as to make the structure obvious. Generally, one would indent all the statements between a THEN, and the end of the action due to the THEN.

```
IF A=B THEN
BEGIN
  ACTION 1
  ACTION 2
  ACTION 3
END;
```

STASM automatically does such formatting, placing a number of "I" characters in front of statements that are indented. The number of "I" characters is equal to the "indent level" of the statement. Since indenting several columns will reduce the room left for comments, STASM moves everything over to the left a bit. Labels do not "stick out" 7 columns in front of the Op Code columns, but are marked with a "I" in front of them. This issue of '68' Micro Journal should contain an article by Derek Gitelson of Sansaska, which includes a listing of his program called DIRE. Refer to that program for the following discussion. Since line numbers are not used in STASM programs (they foul up the format), I will refer to the program lines by the value of the program counter (first columns).

Note line C102 has the label DRVNUM with a "I" in front of it. Next we come to the use of an IF condition on line C119. IF's come in 5 types. This first use is an IFOR type. The original source text uses IFOR on this line and the next. The intent is that if the contents of A equal \$D (a carriage return) OR if the contents of A equal the FLEX EOL character (End Of Line), the action that is described between the word REJECT on line C122, and the ELSEF on line C129 are executed. (I vote for REJECT being changed to THEN, which would make more sense.) If one of the conditions is true, the code between ELSEF and the ENDE on line C186 is skipped. If neither condition is true, the code between REJECT and ELSEF is skipped and the code from ELSEF to ENDE is executed.

The code on line C134 is an example of an IFT. That is, if the condition is true, the code down to the ENDF on line C182 (second one) is executed. IFT signals to execute the code that follows immediately, down to the ENDF at the same level as the initial IFT. Note that the IFT on line C134 is followed immediately by another IFT on line C138. The effect is an AND condition that could be exemplified by some BASIC code:

```
200 IF A=B THEN IF C<3 THEN <ACTION>
```

The code on line C143 exemplifies an IFB code (if branch). The intent is if the contents of A equal \$D, then branch to the label DODIRE. The "L" at the end signifies that a long branch is required. It is up to the programmer to make that decision. Note that STASM removes the distinguishing suffixes from the different IF's, but it is not too difficult to figure out which form is being used by the parameter list for the MACRO. (I would prefer to have the distinguishing part of the IF left on, at least optionally, so I as a beginner could see the distinction more clearly).

Down at line C160 you will find a DO statement. Its action terminates at the ENDO on line C168. A DO statement will execute the statements from the DO to the ENDO until the condition for a BREAKIF contained within the loop is fulfilled. This is the 4th type of IF statement BREAKIF B,EQ,#8 should be fairly obvious. The test is for end of file. As I said previously, BREAK causes the execution to continue on the line after the ENDO. In this case, the JSR PRNFF at line C160. Note that the test doesn't necessarily have to involve an accumulator or register, as in IFT D,NE,#1000. It may just test the condition code register as in line C160 which reads IF EQ.

I hadn't intended to write a complete set of instructions for STASM, and this is not yet complete. I would like to point out one further construct, a count loop. Line C340 contains the instruction LDX #4. This sets up a counter. The next line contains the word COUNT which marks the start of the loop. Line C340 contains the instruction ENDC X. ENDC marks the end of the count loop, and the X indicates which register to decrement. The count loop may use any register or any valid addressing mode to specify the counter. That is, a memory location may be used for the count. The limitation is that the counter is always decremented (ie. not incremented), and always by 1.

Note that the listing was done with macro expansion listing turned off, so that the macro expansions are not shown. That would be the normal way to use STASM. I should say that STASM works in conjunction with the standard TSC assembler, not in place of it.

A word or two about the manual for STASM is in order. When I first received STASM to review, I had a bit of difficulty with the manual because it sort of jumped right into the middle of things. I wrote Derek Gitelson a note with a few suggestions, which he has taken. You will find the manual readable and useful, though brief. Several sample listings are supplied that help the understanding of the macros considerably.

I found the software to work perfectly after a very minor initial difficulty. STASM does not output a nicely formatted listing if the 'M' option is specified in the assembler command line. I sent Sansaska samples of listings that were rather garbled, and Derek told me that he had forgotten to mention the fact that the N option should not be used in the instructions, but that he would add that caution. Without line numbers, the whole system works very nicely. Though the listings it produces look a bit cryptic at first, an hour or so of reading the manual and sample listings, will clear up any difficulties you may have.

If you are interested in more structured programs in assembler, that are easier to read, you can purchase STASM from Sansaska Systems, 3311 Concord Blvd., Concord, CA 94519.

Review by: Ron Anderson - - -

SELECTIVE DIR.CMD

SANSASKA SYSTEMS
3311 Concord Blvd.
Concord CA, 94519

A SELECTIVE DIRECTORY .CMD FOR FLEX*
By Derek Gitelson

The standard FLEX package lacks an informative directory command. It comes with a CAT command which will let you list only the names and extensions of your files. This is the same as the DIR command in CP/M**. CP/M also has a STAT command which will provide additional information about a directory entry, tho not too much more. CP/M has a good (one of very few) feature in that file names may be specified using "wild cards". Wild cards allow you to incompletely name a file and all files that match the completely defined part will be matched. For example if you typed: DIR WS*.* you would get a list of all files starting with WS, and all extension types (WSMAIN.COM, WSOVER1.COM, WSHLP.COM, WSEXAMP.TXT etc.). If you typed: DIR XYZ.* you would get a list of all files named XYZ with all extension types.

To remedy this lack in FLEX I decided to write my own directory program which would provide all the information contained in the directory on each file and would allow me to list only those files I was interested in. The resulting program I call DIRE and the complete listing is shown below. The program is written in STASM09 but the standard assembly form should be obvious to all who wish to use it. The program may be entered and used by the original purchaser of this magazine only. For those of you who don't want to copy it source and object is available on a 5 1/4 inch disk for \$10.00 to cover postage and handling. If you don't want to modify the program out of STASM09, STASM09 and source for DIRE are available for the month of publication for \$49.95 including postage and handling (CA add 6.5% tx) from Sansaska Systems, 3311 Concord Blvd, Concord CA, 94519.

Now about the program. The calling format is defined at the top of the listing so I will not repeat it here. DIRE resides in the system utility command space of FLEX and will not destroy any programs in regular user memory. You may list the directory to your printer by prefixing P, to the DIRE command in the normal FLEX manner. When the directory is sent to the printer a form feed is sent first and every 60 lines after. If you wish to change the number of lines on a page then change the 60 in the IF A,GE,#60 at C1F5 to however many lines you want. When DIRE is invoked it will print a header showing the current date, disk number, name and date the disk was created. The header is repeated on each printed page. Also shown is the number of sectors left on the disk and the start and end address of the free chain. Next column headings as below are printed:

NO. NAME EXT CREATED ATRB START END SIZE

where the headings have the following meanings:

1. NO. - Entry number in the directory.
2. NAME - The file name.
3. EXT - The extension type.
4. CREATED - The date the file was created.
5. ATRB - The protect attributes of the file.
6. START - The first track and sector of the file.
7. END - The last track and sector of the file.
8. SIZE - The number of sectors in the file.

Following the heading is a blank line and all files that match the file specification are printed.

When called the program is initialized to a default file specification of *.* which will match all files in the directory. If your command line contains a specification for the name only, then the default * for the extension is still in effect. If on the other hand you want to list all files with extension .TXT then you must type *.TXT because the * for the name default is cancelled.

After the command line is processed the disk information record is read to provide the header information and the header is printed. Next follows a DO loop at C1E0-C1FB that prints the file information and additional headers, as needed, until the end of the directory. When the end of the directory is encountered the program exits the DO loop does some housekeeping and returns to FLEX.

A series of subroutines follow and each is explained in its comments. At C42D is the routine that matches the input file specification to the current file name and extension in the directory. This module was written as a stand-alone program and can be incorporated, as is, into other programs where you might need such a function.

* FLEX is a trademark of Technical Systems Consultants.

** CP/M is a trademark of Digital Research.

OPT LIS.PAG

2:DIRE - PRINT DIRECTORY OF DISK 12-26-82 TSC ASSEMBLER PAGE 1

```

OPT NOP
000000000000000000000000
8
0 SANSASKA SYSTEMS 0
0 3311 CONCORD BLVD 0
0 CONCORD CA, 94519 0
0 COPYRIGHT 1982 0
0 ALL RIGHTS RESERVED 0
0
000000000000000000000000

```

```

*****
; PRINT A DIRECTORY OF THE DISK ACCORDING TO THE CALL.
; CALL FORMAT: DIRE (DRIVE, FNAME, EXT)
;
; WHERE THE DIRECTORY NUMBER IS 0-3, THE FNAME IS THE
; FILE NAME AND EXT IS THE EXTENSION.
; THE SYNTAX FOR FILE NAME AND EXTENSION IS:
; $NAME$ OR NAME OR NAMES OR .B$ ETC. THE $ MEANS
; MATCH ANYTHING BEFORE THE FOLLOWING STRINGS OR
; ANYTHING AFTER THE PRECEDING STRING AND THE ?
; MEANS MATCH ANY CHARACTER IN THE STRING AT THAT
; LOCATION. FOR EXAMPLE DIRE 1.B$ WILL LIST ALL
; FILES WITH AND EXTENSION THAT BEGINS IN B: BAK,
; BAC, BIN ETC. DIRE FILE?.TXT WILL LIST FILE1,
; FILE2, FILES, FILE4 ETC ALL WITH .TXT EXTENSION.
*****

```

```

CB40 *DIRFCB EQU SYSFCB
00C1 *P60 EQU 60

```

```

C100 ORG PSEUDO
00C1 SETDP P60
C100 00 29 BRA START
C102 01 *DRVNUM FCB 1 ;SKIP OVER DATA
C103 2A 04 *NAME FCC /A/,4 ;DEFAULT TO DRIVE 1
C105 2A 04 *EXTENS FCC /B/,4 ;WILL MATCH ANYTHING
C107 C103 *NAMEPTR FCB NAME ;MATCH ANY EXTENSION
C109 C105 *EXTPTR FCB EXTENS ;POINTS AT NAME IN INBUF
;POINTS AT EXTENSION IN INBUF
C10B *START
; PARSE THE INPUT BUFFER FOR ANY FILE NAME & EXTENSION.
; SET POINTERS TO THEM FOR LATER USE.
C10B 34 00 PSHS DP
C10D 06 C1 LDA DP60
C10E 1F 00 TFR A,DP
C110 06 CC03 LDA TTYDEP ;GET LINES PER SCREEN
C114 34 02 PSHS A ;SAVE IT
C116 06 CC11 LDA LASTPM ;CHAR THAT TERM'D "DIRE"
C119 IF A,EQ,00B IF IT WAS A CR
C11B IF A,EQ,TTYEOL OR FLEX EOL CHAR
C122 REJECT
C124 *SETDEF
C124 06 CC0C LDA WKRDM ;NOTHING SPEC'D, USE DEFAULT
C127 97 02 STA DRVNUM
C129 ELSEF L
C12C 0D CD27 JSR NATCH ;FIRST CHARACTER AFTER "DIRE"
C12F 0D C3E0 JSR CLAS12 ;CLASSIFY INCL & ?
C132 25 F0 *BCS SETDEF ;NON ALFA/NUMERIC, USE DEFAULT
C134 IF A,NE,"0" IF NOT AN 0
C136 IF A,LE,"0"9 IF NUMERIC
C13C 0D 30 JSUBA 00 ;MUST BE DRIVE, REMOVE BIAS
C13E 97 02 *STA DRVNUM
C140 0D CD27 JSR NATCH
C143 IF A,EQ,00D,DODIRE,L IF DRIVE 0 ONLY
C149 IF A,EQ,TTYEOL,DODIRE,L IF DRIVE 0 ONLY
C150 IF A,NE,"0" DRIVE 0 MUST HAVE .
C154 *BADCOM
C154 0D CD3A JSR RSTAT0 ;RESET TO CRT IN CASE PRINTER
C157 0D C36F JSR FSTRNG
C15A 0D 0A 49 0C *FCC 0D,0A,"Illegal file specification.",4
C177 7E CD03 JSR JMP WARMS
C17A *ENDF
C17A 0D CD27 JSR NATCH ;FIRST CHAR OF NAME
C17D 0D C3E0 JSR CLAS12
C180 25 D2 *BCS BADCOM ;SHOULD BE ALFA/NUMERIC
C182 *ENDF
C182 *ENDF
C182 BE CC14 ILDX LBPDM
C185 30 1F LEAX -1,X ;1st FIRST CHAR OF FILE NAME
C187 9F 07 STX NAMEPTR
C189 0D CD3A JSR NATCH ;MOVE TO END OF FILE NAME
C18C 0E CC14 ILDX LBPDM ;1st 2 PAST END OF FILE NAME
C18F C6 04 ILDB DEBT
C191 IF A,EQ,00B IF END OF LINE
C193 IF A,EQ,TTYEOL OR PSEUDO END OF LINE

```

```

C19A *REJECT
C19C E7 04 *STB 1 ;MARK WITH EDT, 2nd END OF LINE
C19E *ELSEF
C1A0 E7 1F *STB -1,X ;MARK FOR COMPARE
C1A2 *ENDE
C1A2 IF A,EQ,"0" IF TERM'D ON 0, THEN EIT NEXT
C1A6 9F 29 *STX EXTPTA ;MARK EXTENSION
C1A8 0D CD3A JSR NATCH ;MOVE TO END OF EXTENSION
C1AB *IF B,GT,07,BADCOM NO MORE THAN 3 LONG
C1A7 0E CC14 ILDX LBPDM
C1B2 C6 04 ILDB DEBT
C1B4 E7 04 *STB 1 ;MARK EOT FOR COMPARE
C1B6 *ENDE
C1B6 *ENDE
C1B6 *DODIRE
C1B6 0E C1A3 LDX 0INFCB ;1st INFORMATION SECTOR FCB
C1B9 06 02 LDB DRVNUM
C1BB E7 03 STB 3,1 ;SET DRIVE 0
C1BD 06 10 LDA 0PENIR
C1BF A7 04 STA 1 ;SET OPEN DISK INFO REQ
C1C1 0D 51 BSR DOFMS2 ;OPEN IT
C1C3 06 07 LDA 0GETIFR
C1C5 A7 04 STA 1
C1C7 0D 40 BSR DOFMS2 ;READ INFO TO FCB
C1C9 0D 72 BSR PRNHOR ;PRINT HEADER
C1CB 0E CB40 LDX 0DIRFCB
C1CE 96 02 LDA DRVNUM ;GET DRIVE NUMBER
C1D0 A7 03 STA 3,1 ;SET IT
C1D2 06 06 LDA 0PENDR
C1D4 A7 04 STA 1
C1D6 0D 39 BSR DOFMS ;OPEN THE DIRECTORY
C1D8 06 07 LDA 0GETIFR
C1DA A7 04 STA 1 ;SET FOR READ DIRECT DATA
C1DC 4F 00 BLA
C1DD 5F 00 CLRB
C1DE 34 06 PSHS D ;FILE NUMBER TO STACK
C1E0 DO ;READ DIRECTORY ENTRIES
C1E0 0D 45 *BSR READNY
C1E2 *BREAKIF B,EQ,00 IF EOF
C1E6 6D 04 *ST 4,X ;TEST FOR NEVER USED
C1E8 *BREAKIF EQ ;NEVER USED. MUST BE END
C1EA 0D C3FE JSR TYPST ;TEST IF NAME MATCHES INPUT
C1ED IF EQ ;IF TYPE TO PRINT
C1EF 0D C313 JSR PRLINE ;PRINT LINE OF DIRECTORY
C1F2 06 CC1A ILDA CURLIN ;CURRENT LINE ON PAGE
C1F5 C1FS IF A,GE,006 IF PAGE FULL
C1F9 0D 42 *BSR PRNHOR ;NEW PAGE & HEADER
C1FB *ENDE
C1FB *ENDE
C1FB ENDO
C1FD 0D C0BE JSR PRNFF ;FINAL FF OR CR/LF
C200 35 06 PULS D ;LOSE FILE 0
C202 35 02 PULS A ;LINES PER SCREEN
C204 07 CC03 STA TTYDEP ;RESTORE IT IN CASE CHANGED
C207 06 FF LDA 0-1
C209 07 C009 STA TTYPAU ;TURN PAUSE BACK ON
C20C 35 06 PULS OP
C20E 7E CD03 JHP WARMS
C211 0E CB40 *DOFMS LDX 0DIRFCB
C214 3F *DOFMS2 CLRB
C215 0D D406 JSR FMS
C218 IF NE ;IF ERROR
C21A E6 01 *LDB 1,X
C21C IF B,NE,00 ;IF NOT EOF ERROR
C220 0D CD3F JSR RPTERR ;REPORT ERROR
C223 7E CD03 JSR JMP WARMS
C226 *ENDE
C226 ENDF
C226 39 RTS
; READ NEXT DIRECTORY ENTRY, SKIP DELETED OR IMUSED.
C227 *READNI
C227 3F CLRB
C228 DO ;READ TILL FIND OK
C228 EE 62 ILDU 2,S ;GET FILE NUMBER
C22A 33 41 LEAU 1,U ;BUMP IT 1
C22C EF 62 STU 2,S ;BACK
C22E 0D E1 BSR DOFMS

```



```

C230      :BREAKIF D,EQ,00      IF DONE
C234 A6 04      :LDA 4.1      FIRST CHAR OF NAME
C236      :PBEAKIF A,GE,01      IF LEGIT NAME
C23A      ENDD
C23E 39      RTS
* PRINT DIRECTORY HEADER AS:
I MM/DD/YY
I DISK 0----, (NAME), CREATED: MM/DD/YY
I SECTORS LEFT: ---- (START: ----; END: ----)
I NO. NAME E11 CREATED ATTR START END SIZE
I .....
C23D      *PRINHDR
C23D 0D C38E      JSR PRNFF      START WITH CLEAN PAGE
C240 0E C08E      LDI 0MONTH
C243 0D C387      JSR DATEOUT      PRINT TODAY'S DATE
C246 0D C024      JSR PCRLF
C249 0D C36F      JSR FSTRING
C24C 44 49 53 0D      FCC 'DISK 0',4
C253 0E C4FE      LDI 0INFORMCB+050      DISK 0
C256 5F          CLRB      NO LEADING SPACES
C257 0D C039      JSR OUTDEC      PRINT DISK 0
C25A 0D C36F      JSR FSTRING
C25D 2C 20 20 04      FCC ' ', ' ',4
C261 0E C4F3      LDI 0INFORMCB+050      DISK NAME
C264 C6 00          LDB 011
C266 0D C398      JSR ISTRING      PRINT DISK NAME
C269 0D C36F      JSR FSTRING
C26C 2C 20 20 43      FCC ' ', CREATED: ',4
C279 0E C506      LDI 0INFORMCB+063      INITIALIZATION DATE
C27C 0D C387      JSR DATEOUT
C27F 0D C024      JSR PCRLF
C282 0D C36F      JSR FSTRING
C285 53 45 43 54      FCC 'SECTORS LEFT: ',4
C294 0E C504      LDI 0INFORMCB+061      FREE SECTORS
C297 5F          CLPB
C298 0D C039      JSR OUTDEC
C29B 0D C36F      JSR FSTRING
C29E 2C 20 20 53      FCC ' ', (START: ',4
C2A9 0E C500      LDI 0INFORMCB+05D      START OF FREE CHAIN
C2AC 5F          CLRB
C2AD 0D C03C      JSR OUTHEX
C2B0 30 01          INX
C2B2 0D C03C      JSR OUTHEX
C2B5 0D C36F      JSR FSTRING
C2B8 3D 20 20 45      FCC ' ', END: ',4
C2C1 0E C502      LDI 0INFORMCB+05F      END OF FREE CHAIN
C2C4 0D C03C      JSR OUTHEX
C2C7 30 01          INI
C2C9 0D C03C      JSR OUTHEX
C2CC 86 29          LDA 0'1
C2CE 0D C018      JSR PUTCHR
C2D1 0D C024      JSR PCRLF
C2D4 0D C36F      JSR FSTRING
C2D7 20 4E 4F 2E      FCC ' NO. NAME EXT'
C2EB 20 20 43 52      FCC ' CREATED ATTR START END SIZE'
C30E 04          FCB 001
C30F 0D C024      JSR PCRLF
C312 39      RTS
* PRINT LINE OF DATA
C313      *PRLINE
C313 0D C024      JSR PCRLF
C316 30 62          LEAT 2,5      POINT AT FILE COUNT
C318 C6 01          LDB 01      PRINT LEAD SPACES
C31A 0D C039      JSR OUTDEC      PRINT FILE NUMBER
C31B 0D 5F          BSR SPACE2
C31F C6 00          LDB 00
C321 0E C044      LDI 0DIRFCB+4      FILE NAME
C324 0D C398      JSR ISTRING      PRINT FILE NAME OF 8 CHAR
C327 0D C37E      JSR SPACE2      PRINT 2 SPACES
C32A C6 03          LDB 03
C32C 0D C398      JSR ISTRING
C32F 0D C37E      JSR SPACE2
C332 0E C039      LDI 0DIRFCB+25      DATE OF CREATION
C333 0D C387      JSR DATEOUT      PRINT IT TOO
C338 0D 44          BSR SPACE2
C33A F6 C04F      LDB 0DIRFCB+15      ATTRIBUTES
C33B CE C368      LDB 0WDR      A1 FLAGS
C340 0E 0004      LDI 04      COUNTER
C343          COUNT
C345 A6 C0          :LDA 0+
C345 38          :LDB 0

```

```

C346      :IF CC      IF OFF
C348 06 20          :LDA 0020      ATTRIB NOT PRESENT. DON'T PRINT
C34A      :ENDF
C34A 0D C018      JSR PUTCHR      PRINT ATTRIB
C34D      ENDC 1
C351 0D 20          BSR SPACE2
C353 0E C051      LDI 0DIRFCB+17
C356 0D C045      JSR OUTADR      PRINT FIRST SECTOR NUMBER
C359 0D 23          BSR SPACE2
C35B 3D 01          INI
C35D 0D C045      JSR OUTADR      PRINT END SECTOR NUMBER
C360 0D C37E      JSR SPACE2
C363 3D 01          INI
C365 C6 01          LDB 01      PRINT LEADING SPACES
C367 0D C039      JSR OUTDEC      PRINT SIZE
C36A 39      RTS
C36B      *WDR
C36B 57 44 52 43      FCC 'WDR'
* PRINT STRING FOLLOWING JSR
I USING FILE FSTRING
*FSTRING
C36F      *FSTRING
C36F 35 10          PULS X      *1 AT STRING
C371          DO      PRINT IT
C371 A6 00          :LDA 1+
C373      :BREAKIF A,EQ,0E0T      IF END
C377 0D C018      JSR PUTCHR
C37A      ENDD
C37C 1F 15          TFM X,PC      RETURN TO PROGRAM
I PRINT 2 SPACES
*SPACE2
C37E      *SPACE2
C37E 06 20          LDA 0020
C380 0D C018      JSR PUTCHR
C383 0D C018      JSR PUTCHR
C386 39      RTS
I PRINT DATE IN MM/DD/YY FORMAT.
I 1 MONTH ON ENTRY.
C387      *DATEOUT
C387 0D 1C          BSR DECIDUT      PRINT MONTH
C389 0D 07          BSR SLASH      PRINT SLASH
C38B 0D 18          BSR DECIDUT      PRINT DAY
C38D 0D 03          BSR SLASH
C38F 0D 14          BSR DECIDUT      PRINT YEAR
C391 39      RTS
* PRINT A /
*SLASH
C392      *SLASH
C392 06 2F          LDA 0'1
C394 0D C018      JSR PUTCHR
C397 39      RTS
I PRINT STRING STARTING AT 'X, (01 CHAR LONG.
I USE SPACES FOR NULLS.
C398      *FSTRING
C398      COUNT
C398 A6 00          :LDA 1+      GET CHAR TO PRINT
C39A      :IF EQ      IF NULL
C39C 06 20          :LDA 0020      USE SPACE
C39E      :ENDF
C39E 0D C018      JSR PUTCHR
C3A1      ENDC 0
C3A4 39      RTS
I ONE BYTE DECIMAL OUTPUT ROUTINE. PRINT LEADING 0'S
*DECIDUT
C3A5      *DECIDUT
C3A5 E6 00          LDB 1+
C3A7 4F          CLRB
C3AB          DO
C3AB 4C          :INCA
C3AB C0 0A          :SUBB 010      COUNT 10'S
C3AD          :BREAKIF N1      SUB IT NOW
C3AD      ENDD      IF TO FAR
C3AD 0D 0A          ADDA 0'0-1
C3AF 0D C018      JSR PUTCHR      REPLACE IT
C3B0 1F 00          TFR 0,A      UNCOUNT & MAKE ASC11
C3B2 0D 30          ADDA 0'0      PRINT IT
C3B4 0D C018      JSR PUTCHR      MAKE LEFT OVER ASC11
C3B7 39      RTS
C3BE      *PRNFF

```

``` # PRINT FORM FEED ```

```
C30E FC C010 LDB OUTCH+1      OUTPUT VECTOR
C3C1      IF D,ME,OUTCH2+1  IF NOT TO CRT
C3C7 B0 C36F JSR FSTRNG      PRINT NEXT
C3CA 0C 04      FCR 0C,4      A FORM FEED
E3CC B6 3D      LDA 061
C3CE B7 C003 STA ITYPEP      SET LARGE PAGE SIZE
C3D1      ELSEF      TO CRT
C3D3 B0 C024 JSR PCRLF
C3D6 B0 C024 JSR PCRLF
C3D9      ENDE
C3D9 39      RTS

C3DA      *NITAM
# SCAN THE INPUT BUFFER FOR THE NEXT TERM CHAR
# (NON ALFA/NUMERIC/0/?) AND RETURN IT IN A.
# ALSO COUNT NUMBER OF GOOD CHARACTERS

C3DE      IF CS      IF NOT ALFA/NUMERIC BY FILE
C3E0 B0 09      BSR CLAS12  TRY NY DEF
C3E2      BSRCLAS12
C3E4 5C      INCB      IF STILL NOT ALFA/NUMERIC
C3E5      ENDF      COUNT ALFA/NUMERIC
C3E5      ENDO
C3E7 39      RTS

C3E8      *CLASS1
# RETURN CC IF (A) IS 0 OR ?

C3EB B0 C021 JSR CLASS      FIRST TEST NORMAL
C3EB      *CLASS2
C3ED      IF CS
C3ED      IF A,EQ,"0"0"      IF AM 1
C3F1      IF A,EQ,"0"?"      OR A ?
C3F3      REJECT
C3F7 1C FE      AMBCC 00FE      CLEAR CARRY CAUSE ALFA/NUMERIC
C3F9      ELSEF
C3FB 1A 01      ORCC 001      SET CARRY CAUSE NOT ALFA/NUMERIC
C3FB      ENDE
C3FB      ENDF
C3FD 39      RTS
C3FE      *TYPTST
# RETURN EQ IF CURRENT FILE MATCHES CALLING SPEC.

C3FE 31 04      LEAY 4,Y      Y= FILE NAME
C400 7E 07      LDI NAMPTR      1= TARGET
C402 1F 23      TFR Y,U      MEASURE FILE NAME LENGTH
C404 5F      CLRB
C405      DO
C405 A6 C0      LDA U+
C407      BSR COMPARE      CHECK FOR MATCH
C409 5C      INCB      COUNT FILE NAME
C40A      BSRCLAS12      MAX 8 CHAR LONG
C40E      ENDO
C410 B0 10      BSR COMPARE      CHECK FOR MATCH
C412      IF ED      IF NAME MATCHES
C414 7E 09      LDI EXTPIR      1= TARGET EXTENSION
C416 10BE C04C      LDI QDIRPCD+12  Y= EXTENSION
C41A 1F 23      TFR Y,U      COUNT EXTENSION
C41C 5F      CLRB
C41D      DO
C41D A6 C0      LDA U+
C41F B9 C7      BSR CLASS1
C421      BSRCLAS12      NOT LEGIT
C423      BSRCLAS12      3 CHAR MAX
C427 5C      INCB
C428      ENDO
C42A B0 01      BSR COMPARE      CHECK IT
C42C      ENDF      RETURNS EQ/NE
C42E 39      RTS
```

```
*****
#
# Compare two strings of length (B). The string
# 1 is the target and the one 2 is the one to
# search. If the first or last character of the
# target is an 0 then any number of non-matching
# characters can precede or follow, respectively,
# the target string. If any character in the target
# is a ? then any character in the string 2 will
# match it.
```

```
#
# Enter with: B = Length of string to search
#            1 = Target string, terminated w/EOT=4
#            Y = String to search
#
# Returns: A,B=?
#           U,1,Y=U,1,Y =
#           1=1: match, 2 = 0: no match
*****

0000 *TARGLN EQU # Length of target string
0001 *NDSERST EQU TARGLN+1 end of string to search
0003 *WSPAC EQU NDSERST+2 end of workspace
0005 *LENGTH EQU WSPAC+0 Length of search string on stack
0006 *TARGET EQU LENGTH+1 1st target string
0006 *SERSTR EQU TARGET+2 start of string to search

# Equates

C420      *COMPARE
C420 34 74      PSMS 0,1,Y,U
C42F 32 7D      LEAB -WSPAC,S      room to work
C431 1F 43      TFR S,U
C433 5F      CLRB      measure target length
C434      DO
C434 A6 B0      LDA 1+
C436      BSRCLAS12      if end
C43A      IF A,NE,"0"0"      count non 0 characters
C43E 5C      INCB
C43F      ENDF
C43F      ENDO
C441 E7 C4      STB TARGLN,U      set length of target
C443      IF NE      if any target
C445 1F 20      TFR Y,U
C447 E0 43      ADDB LENGTH,U      0= end of string to search
C449      IF CS
C44B 4C      INCB
C44C      ENDF
C44C E0 41      STD NDSERST,U
C44E AE 44      LDI TARGET,U      1= target string
C450 A6 B4      LDA 1
C452      IF A,EQ,"0"0"      if on 0
C456 30 01      LEA1 1,1
C458      DO      scan for match
C458 34 30      PSMS 1,Y
C45A 1F 20      TFR Y,U      test that endl left to match
C45C E0 C4      ADDB TARGLN,U
C45E      IF CS
C460 4C      INCB
C461      ENDF
C461      BSRCLAS12      search string(target
C466 B0 22      BSR MATCH      room to match
C468      IF ED      if possible match
C46A B0 33      BSR NOTEST      verify end of search string
C46C      BSRCLAS12      regardless
C46E      ENDF
C46E      BSRCLAS12      if partial match ok
C472 35 30      PULS 1,Y
C474 31 21      LEAY 1,Y      move up search string
C476      ENDO
C478 32 64      LEAS 0,S      clear 1 & Y from stack
C47A      ELSEF      partial at start not OK
C47C B0 0C      BSR MATCH
C47E      IF NE      if no match
C480 B1 2A      CMPA 0"1
C482      ELSEF      if end on partial OK
C484 B0 19      BSR NOTEST      verify end of search string
C486      ENDF
C486      ENDF
C486 32 64      LEAS WSPAC+1,S      clear workspace & lose B
C488 35 F0      PULS 1,Y,U,PC
C48A      *MATCH
# Match strings 1 and 2. Return 1=1 if match up to
# EOT in 1. If ? in 1 string then it matches all.
# Return character of 1 in A. Return 2=0 if no match.

C48A      DO
C48A A6 B0      LDA 1+      target char
C48C      BSRCLAS12      & sets 1=1
C48E      IF A,EQ,"0"?"      2=0
C494 31 21      LEAY 1,Y      match anything in 2,Y.
```

```

C496      :ELSEF
C498      ::BREAKIF 4,HE,V+      if not match
C49C      :ENDE
C49C      ENDO
C49E 39   RTS

C49F      +NDTEST
          ! Test and be sure at end of string to search

C49F 10AC 41      CMPY 40SERST,U      Z=1 AT END OF SEARCH STRING
C4A2 39          RTS
C4A3      +INFOC0 RMB      340
          END START

```

XADR

ACORN XADR & XMPR

In the past month we have received two additional products from ACORN Computer Systems for review. We will review the XADR board here and the XMPR (SWTPC MPR Programmer extender for 2732-2764) at a later date.

XADR

Extended Addressing for older S50 Computers

The evolution of the Standard S50 Bus computers has drastically changed the memory addressing scheme used originally to what is available today. Mainly the advent of compacted memory ICs that allow much more memory available per chip count.

Seems just like yesterday that the 2101 and 2102 were real winners. Each was in it's day, but that day has fast receding, and we now can purchase memory chips with 8 and 16 or more times as much memory in the same amount of board space. Hence, the availability of more RAM than most 8 bit microprocessors can address. This has brought about different methods of housekeeping a larger array of memory and addressing it. As to the housekeeping chores, various DAT (data address translator) programs were installed on microprocessors to read and write to this expanded memory resource. Among the very earliest was the SWTPC DAT as well as others. This allowed the entire memory base to be addressed, either as a logical or physical address space, up to about a megabyte or so. But then many found that they could not address this additional memory as their system did not have the additional memory addressing lines (A16..A19), this applies to most all pre-1980 Standard S50 Bus computers.

Well for those who have the older computer the ACORN XADR extended address board can be the solution. The XADR can upgrade the older motherboards to the current S50C standard, with the additional address lines available. In most cases the baud rate components are removed from the CPU card and installed on the XADR, this frees up the extended address lines (the additional lines, for extended addressing, are where your baud rate lines are now). If your motherboard already has the baud rate components installed then this is not necessary.

The XADR also provides a programmable, buffered parallel port (MC6821) with 4 buffered handshaking lines. The port may be changed from output to input by switching one IC from one socket to another. This I/O port is the 'A' side of the PIA.

DAT

In developing this for their own system, ACORN personnel considered most DAT methods of extended addressing was too complicated for single-ended users. Also they stated that they felt it was slow in data transfers between PROM and RAM memory boards. They have developed their entire system around the fact that most single-ended users would be using the additional memory as either pseudo-disk space, for rapid access, or to store, in EPROM those often used commands normally stored on the system disk.

With the falling cost of RAM and the every expanding size of data files, this appears to be a valid assumption. It essentially turns your older computer into a 904K (940,000 byte) virtual disk computer!

It can be thought of as a large disk with 16 tracks (extended addressing pages) and up to 224 sectors (56K bytes) per track. The top 8K is used for ports, video RAM, monitor and PRAM-TRANS (reviewed in an earlier issue of 68 MICRO JOURNAL) software and is available to all pages. PRAM-TRANS is a software package that loads

FLEX™ .CMD from PROM. It is called through the user look-up table in FLEX, transferring programs from PROM to RAM.

Example, by using the ACORN 168K PROM DISK board you can eliminate 'booting' FLEX and also store, in addition to FLEX, all standard FLEX commands, Editor, Assembler, XBASIC and XPC, using about or less than 90K bytes. This leaves you (using one PROM DISK board) about 80K bytes of useable data RAM. Up to four of these boards can be installed in your system.

Timing

Normally it takes about 14 seconds to load XBASIC from a 5" disk (appx. 20K bytes). Using the PRAM-TRANS utility the same program will be loaded from EPROM to RAM as follows (approximate times due to different xtal tolerance): 1 MHZ 6800 - .37 seconds, 2 MHZ 6800 - .19 seconds, 1 MHZ 6809 - .2 seconds, 2 MHZ 6809 - .1 seconds. Loading and chaining XBASIC programs will not be quite this fast but should be 10-20 times faster!

The PIA Port

ACORN uses a unique method to control the PIA for extended addressing. They do it with the R/W line. Bits (0-3) are the external address for READING, and bits (4-7) are the external address for WRITING. Storing an \$F in the PIA will READ from page \$F and write the data to page \$0. Clearing the PIA will return the computer to a normal state after a transfer. The 'B' side of the PIA is used for extended addressing.

Additional Software

The following additional software packages are either available or being developed for this system:

1. PRAM-TRANS (see above)
2. PRAM-DISK Saves and loads TEXT, DATA and BASIC from RAM or EPROM. It requires a special set of disk drivers for drive #3, and operates through the normal FLEX routines.
3. RAM-DISK Virtual memory disk, for use with the extended RAM boards. Extended RAM is formatted as a disk and is accessed by modified drivers as drive #3.
4. PRAM-FORMATTER Formats files and updates the directory for burning into EPROMS.

Conclusions

The XADR is under warranty for 90 from date of purchase for all assembled and tested boards sold. Kits have a 90 days parts warranty. Like other ACORN products used and reviewed, this system is well designed and the documentation is more than sufficient. We feel that it is the practical answer to users of older Standard S50 Bus computers, and increases the utility of any non S50C 68XX computer.

Additional information can be obtained by contacting:

ACORN Computer Systems
11931 W. Bluemound Road
Milwaukee, WI 53226
(414) 257-0300

CONDUIT

A Review of CONDUIT

by E. M. (Bud) Pass, Ph.D.
Computer Systems Consultants, Inc.
1454 Latta Lane, Conyers, GA 30207
Telephone Number 404-483-1717/4570

GENERAL

CONDUIT is a set of modules for OS9 which generalizes the concept of pipelines. It

consists of file manager, device descriptor, device driver, and utility modules. It can be used in most existing programs wherever a device independent sequential file would be used. Also, new programs can be written to take advantage of the additional power of conduits.

Pipelines, for the uninformed, are buffered connections from one module to another. One module outputs to the pipeline and the other inputs from it. Other than the fact that the modules are run in multitasking mode, the result of a pipeline is as if the first module were run to completion, outputting its results to a disk file, followed by the execution of the second module, inputting from that file. An OS9 pipeline connecting two modules is represented by a "!" separating the module names on a command line. The module on the left is expected to send output to its STDOUT file (path 1) and the module on the right is expected to receive this data from its STDIN file (path 0). For example the following command line specifies a pipeline from MODONE to MODTWO:

```
MODONE ! MODTWO
```

COMPARISON

In their simplest form, conduits may be viewed as bidirectional pipelines. The following table briefly compares pipelines and conduits:

Pipelines	Conduits
Invoked by SHELL with delimiter "!" on command line	May be invoked by module as a normal pathlist
Left to right	Bidirectional
Limited to redirectable paths (0/1)	Usable as a normal path
Sequential transfers only	Sequential or random transfers
Device and file independent only	All file types

The conduit device is named "c", and individual conduits are named "/c/pathname", where "pathname" is used to connect the files at each end of the

conduit. Thus, a conduit connecting modules MODONE and MODTWO could be represented symbolically as follows:

```
MODONE >/c/condx & MODTWO </c/condx
```

CONDUITS

Conduits have the capability of being used for certain purposes for which they seem ideally suited and provide elegant solutions to complex problems.

They may be used to connect multiple files in two modules in a unidirectional fashion which resembles pipelines. This has the same advantages as pipelining the data, and is simply a generalization of pipelines. It is the easiest to implement, requiring no changes in the programs except to change the corresponding file names to conduit names.

They may be used to create filters, rings, networks, and other exotic multi-module structures which may simplify overall system design by allowing each module to be designed for one function, and yet allowing each module to communicate bidirectionally with potentially all other modules in the structure.

They may be used to simulate terminal input and output with a canned program to accomplish complicated functions without the necessity of writing custom programs or rekeying command sequences. The PRETTY utility illustrates this capability.

Several utility programs are provided in the CONDUIT package, some of which are directly related to the remainder of the package and some of which are not.

The first utility is called DUP. It is a replacement for the COPY utility and offers the following advantages over it:

- reuses old output space allocations,
- defers verification until end,
- optionally omits verification.

Another utility is called PRETTY. It runs a BASIC09 session in a terminal simulation mode. It submits BASIC09 programs for compilation, captures the indented listing, and enhances it with headings and pagination. It is an example of the use of conduits with

canned programs for dedicated usage.

Still another utility is called TURNAROUND. It echoes its standard input file to its standard output file, closing its output file in response to its input file being closed. It may be used to great advantage with conduits in such applications as allowing BASIC09 programs to perform complex data formatting and conversions by outputting to one file and immediately reading the outputted data from another file.

Another utility is NULL. It provides a means of running a program which normally outputs to a printer or disk file when it is desired not to create the file. This capability is called "DUMMY" on IBM mainframes and "BLACKHOLE" on certain minicomputers.

An additional utility is REDIRECT. It intercepts all traffic between the OS9 SHELL and the terminal. It provides means of dynamically redirecting the standard paths STDIN, STDOUT, and STDERR without invoking a new copy of SHELL.

Still another utility is TUCK. It saves space by packing multiple OS9 modules into one 1K (or other size) space; ordinarily OS9 modules start at 256 byte boundaries.

The last set of utilities described is actually a group of macros for the Microware OS9 editor. They perform such activities with conduits as providing faster file saving operations using DUP, executing the assembler from the editor, etc.

ADVANCED CAPABILITIES

The CONDUIT manual suggests several advanced applications of conduits which are possible using programs explicitly written using conduits.

It is possible to monitor data streams in search of control sequences used to trigger character substitution or subroutine execution.

It is possible to write scripts for demonstrations of software packages to exercise various capabilities of interest to different classes of customers without modifying the software packages in many cases.

It is possible to add features to other file types, such as read backward sequential to RBF access.

It is possible to create complex data types, such as indexed sequential access, which appear like simple data types to other modules.

It is possible to map one device into another. Thus a system supporting multiple types of terminals could be written using one type of terminal in all programs and multiple screen mapping conduit programs. Also a single logical disk file could be mapped into multiple physical disk drives.

It is possible to write a message switching system by providing a conduit program for each logical terminal in a network. In this case, each logical terminal would have a name, which would begin every message directed to it. Also, logical terminals could easily be assigned to disk files, printers, alternate terminals, modems, etc. without modifying any of the programs supporting the message switching system.

CONDUIT MANUAL AND DISKETTE

The CONDUIT manual is very thorough as a reference manual. It provides detailed discussions of how to use conduits in various modes of operation. It offers many suggestions for the use of conduits in general, providing several reasonably detailed examples. It discusses the causes and cures of many types of problems commonly encountered in the use of CONDUIT, especially with respect to error handling. It covers the utilities provided in the CONDUIT package and their potential uses. It suggests several means of installing the requisite modules composing CONDUIT, which occupies about 1.5K bytes total, not including any of the utilities.

The CONDUIT diskette contains the following files:

conduit	conduit modules
cond_macros	edit macro sources
dup_asm	utility source
pretty-basic	utility source
turnaround_asm	utility source
null_asm	utility source
redirect_asm	utility source

tuck_asm	utility source
dup	utility object
pretty	utility i-code
turnaround	utility object
null	utility object
redirect	utility object
tuck	utility object
start_redirect	SHELL command file
change_name	SHELL command file
fm_fixup	SHELL command file
pass_codes	map of status codes

COMMENTS

The CONDUIT package seems to be well documented and well thought out. The examples are adequate and provide a base for understanding the fundamentals of the usage of conduits for an OS9 expert.

Unfortunately, the CONDUIT manual suffers from the same problems as do OS9 manuals. They are all reference manuals, and much of the material being covered is far too complex for the beginner or semi-expert to comprehend without tremendous effort. In order to understand many of the simpler concepts of pipelines and conduits, a manual the size of the current one (about 50 pages) would be required. Then some of the more complex concepts, such as dynamic memory allocation and multitasking, could be covered in a still longer manual. The user then might be ready for the reference manual.

There is a potentially serious problem with the use of conduits on OS9 Level 1 due to memory size restrictions. In the complex applications contemplated by the designers of CONDUIT, great demands will be placed on the memory space available. Because of the very real potential of memory address space fragmentation, the dynamic loading of modules will not be reliable enough or fast enough for many applications. Hopefully, the implementation of conduits on OS9 Level 2 will support multiple address spaces, providing an effective means of using more than 64K bytes of memory for a network of cooperating modules connected by conduits.

SUMMARY

If you believe you have the need for conduits, the price is right. The manual is available for \$8.00, the program package disk is available for \$70.00, and

both together are available for \$75.00, plus NY sales tax or foreign shipping for 0.25 kgm. The name and address of the company are as follows:

MACROPLEX Software
175 Fifth Avenue
Suite 3011
New York, NY 10010

This version is for OS9 Level 1 (version 1.2 or above). A version for OS9 Level 2 is promised but has not been reviewed.

PURCHASE ORDER WRITER

By: R.O. 'Slim' Cummings
M & S SYSTEMS
PO BOX 187
Pittsburg, KS 66762-0187
(316) 232-2264

Ordering and keeping track of those orders can be a big job. The following FLEX-XBASIC program provides a disk record of all orders as well as a file of names and addresses eliminating the necessity to look-up and re-enter them.

The program is written in TSC BASIC pre-compiled format. I am really convinced that this is the only way to fly. The process of edit-compile-run is a pain when debugging, but the completed program is readable in the future for revision.

The program first initializes itself including revising the DATE\$ information into a more useful form. The month information is available as integer-MO\$, as a string(numbers)-MONTH\$, and as a string(alpha)-MONTH \$.

Part of this information is used to generate a filename and a purchase order number for the current order. They are generated by concatenating the first three letters of the company name with the number format day, month and year. (ie. AC026023 for Acorn Computer Systems order of 26 January, 1983.)

I believe the program is self-prompting as it is run. Only the letter (upper or lower case) in parenthesis need be entered - no return. A new order may be (I)nitialiated, it may be (C)hecked, (F)illed, (P)rinted, or an (E)nvelope printed. An old order may be called up by (L)oadung it and any of the above functions run. Anytime the (I)nitialiated portion is run and the company name is requested, the disk is searched for a company with the same first 5 characters. The company name is printed and if the prompt is answered as (Y)es, that file information is used. If not, the file is searched to its end for the proper company name. If the correct company name is not found, a prompt is issued to input a new address. That address is entered into the address file.

Order data is requested - quantity, description, unit prices. If an additional line is needed for the description, a negative quantity will generate a line without quantity or price information. If the item is priced as 3 for \$1.00, the unit price may be input as 3/1 and the unit price will be calculated.

Input is terminated by entering 0 as the quantity. Shipping charges are requested and may be input as an absolute price or percentage. If the number entered is followed immediately by a % the shipping charge will be calculated from the total.

Method of payment follows and the ones included here are (V)isa, (M)aster Card, (C)heck, (B)ill me. The statement to be printed along with the credit card numbers are in the PRINT Subroutine, near the end. The name to be printed at the bottom of the page is also in this portion of the program.

Finally any comments to be appended to the order are requested. The menu is again printed, and the data may be (C)hecked. The check routine prints the P.O. as it will appear, except comments (if any) are omitted. A correction is requested simply by typing the number of the item to be revised, the information must then be reentered in the same - quantity, description, price - format.

When the order is complete it can be (F)illed. The proposed filename is printed and input is requested as to its validity. If there is an existing file with that filename, it will be deleted and the new one will replace it. This may be avoided by entering a new filename.

(P)rint simply outputs the data to the printer or to an out file and returns to the menu.

(E)nvelope prints the address information on an appropriate place on the envelope.

(L)oad prompts for the appropriate filename (purchase order number or filename will work). If the P.O.# is unknown, a "?" may be input. A check of the directory can then be done and the program restarted with 'RUN'.

(Q)uit exits the program to XBASIC.

NOTE: No provision has been made for the first address to initiate the ADDRESS.OAT file. It must be generated with at least one address via the editor before the program is run. The address file is then self generating after this first entry. (A sample ADDRESS.DAT file is included after the program listing. The first line of the file must be the number of entries in the file. The next line is the company name, address, additional line of address (or 0 if none is needed). The last line is City, State, ZIP. These two lines are repeated for all addresses. This file is kept in alphabetical order by the program and this order must be preserved, as the binary search expects it.

This program has been very useful to me, it provides a record of past purchase by date and vendor as well as an address file to slim my new orders.

```
*****
* Purchase Order Writer *
* 1st P.O. "Silo" Company 2/19/81 *
* M & S Systems *
* P.O. Box 107 *
* Pittsburg, KS 64762 *
* *
* Last rev. 26 Feb. 1981 *
*****
```

```
OR EPOD GOTO EPR_RDU
DIR QTY(100),ITEM(100),UNIT PRICE(100)
DIM CO_NAME(50),CO_ADDR(50),CO_CITY(50),
CO_STATE(50),CO_ZIP(50)
MOVES=CHG(112): PER Move & Clear for terminal used.
TOTAL = 0.0
PRINT EPR="1"
SLASH = "/"
PAGE20=" "
PAGE21=" "
PAGE22=" "
```

• Create data parameters

```
MON = 0:DAT(1),2,3
FOR MONTH = 1 TO 12
  READ MONTH_9,MON_9
  IF MONTH = MONTH_9 THEN MONTH_9 = MON_9 : GOTO START
NEXT MONTH
DATA JAN,January,FEB,February,MAR,March,APR,April,MAY,May,1
JUN,June,JUL,July,AUG,August,SEP,September,1
OCT,October,NOV,November,DEC,December
START IF MON < 10 THEN MONTH = "0"&(100-STR$(MONTH_9,2,1)) : GOTO DAY
MONTH = MID$(STR$(MONTH_9,2,1))
DAY = LEFT$(DATE$,2)
YEAR = MID$(DATE$,2)
MENU PRINT MOVES: FOR I=0 TO 5 : PRINT : NEXT I
PRINT " Do you want to: "
PRINT " (Q)uit "
PRINT " (I)nitiate an order "
PRINT " (C)heck order "
PRINT " (P)rint order "
PRINT " (F)ile "
PRINT " (E)nvelope Print "
PRINT " (L)oad "
```

* (The expression in the following line (CHR\$(ASC(112))) AND 223))
* Inputs a single character from the keyboard and converts it to upper
* case if it is lower case. The converted character will be a control

```
* character if a character less than 0 is entered.)
MENU_9 = CHR$(ASC(112)) AND 223: PRINT
IF MENU_9 = "Q" THEN END
IF MENU_9 = "I" THEN GOSUB INIT_
IF MENU_9 = "C" THEN GOSUB CHECK_
IF MENU_9 = "P" THEN GOSUB PRINT_
IF MENU_9 = "F" THEN GOSUB FILE_
IF MENU_9 = "L" THEN GOSUB LOAD_
IF MENU_9 = "E" THEN GOSUB ENVEL_
GOTO MENU
```

• Initiate P.O.

```
INIT_ INPUT "Company Name "CO_NAME
GOSUB OSK_ADR
IF CO_9 = "1" GOTO P_O_NUM
INPUT "Company Address "CO_ADDR
INPUT "Company Address (0 if none) "CO_ADDR16
INPUT "Company City,St,Zip "CO_CITY,CO_ST,CO_ZIP
* (P.O. Number and Filename will both be the first three characters
* of the company name and the date (i.e. 11200000)
P_O_NUM P_O_NUM = LEFT$(CO_NAME,3)&DATE$&MONTH$&DAY$&YEAR$,11
FILE_9 = P_O_NUM
IF CO_9 = "1" THEN GOSUB OSK_INIT
FOR I=1 TO 24 : PRINT : NEXT I
PRINT CO_NAME
PRINT CO_ADDR
IF CO_ADDR16 < "0" THEN PRINT CO_ADDR16
PRINT CO_CITY, "CO_ST": "CO_ZIP
PRINT : PRINT P_O_NUM : PRINT
```

• Input Order Data

```
IMP_DAT PRINT "Input 0 to Terminate entries"
FOR I=1 TO 100
  GOSUB IN_ITEM
  IF OTT(I) = 0 GOTO DATA1
NEXT I
DATA1 INPUT "Shipping Charge 11 Percent - Input Percentage + 1) "SHIPPING6
PRINT "Method of Payment-(V)isa,(R)C,(C)heck,(D)ill oo."
PAY_MTH = CHR$(ASC(112)) AND 223: PRINT
PRINT "Input any Comments to be appended (0 if none) "
INPUT LINE COMMENT$
INPUT "Discount 1 (input 0 if none)"DISC0
RETURN
```

```
IN_ITEM PRINT "Item 0 -"11
INPUT "Quantity -"10TY(I)
IF OTY(I) = 0 THEN RETURN
IN_1 PRINT "Description-"1
INPUT LINE ITEM(I)
IF LEN$(ITEM(I)) > 62 THEN PRINT
"Line too long - Additional line possible - enter negative Qty."
GOTO IN_1
IF OTY(I) < 0 THEN UNIT PRICE(I) = 0 : RETURN
INPUT "Unit Price -"UNIT PRICE
SLASH_LOC = INSTR(1,UNIT PRICE,SLASH)
IF SLASH_LOC = 0 THEN UNIT PRICE(I) = VAL(UNIT PRICE) : RETURN
PRICE_ = VAL(MID$(UNIT PRICE,SLASH_LOC+1))
OTT_ = VAL(LEFT$(UNIT PRICE,SLASH_LOC-1))
UNIT PRICE(I) = PRICE_&OTT_
RETURN
```

• Check & repair Subroutine

```
CHECK_ PRINTER="1"
GOSUB PRIN_1
PRINT "Do you want to revise any data Y/N ?(Y=11200000:PRINT
CONT IF DS = "Y" OR DS = "y" THEN INPUT "Which item 0-100 for Shipping?"112:1
IF I1 > 200 THEN GOTO ED_SHIP
ELSE GOSUB IN_ITEM:1
PRINT "Another item 0? "1:DS = 11200000:PRINT:1
GOTO CONT
RETURN
```

```
ED_SHIP INPUT "Shipping Charge 11 Percent - Input Percent + 11" SHIPPING6
RETURN
```

```
ENVEL_ PRINT "Insert Envelope & Type Return":CO_9 = 11200000:PRINT
GOSUB OPR_PTH
PRINT 00, TAB(10)CO_NAME
PRINT 00, TAB(10)CO_ADDR
IF CO_ADDR16 < "0" THEN PRINT 00, TAB(10)CO_ADDR16
PRINT 00, TAB(10)CO_CITY, "CO_ST": "CO_ZIP
CLOSE 0 : PRINTER="1"
RETURN
```

• Print Subroutine

```
PRINT_ GOSUB OPR_PTH
PRINT_1 PRINT 00, TAB(33)100000 "MONTH_0", 10:1YEAR0
PRINT 00,
PRINT 00, CO_NAME
PRINT 00, CO_ADDR
IF CO_ADDR16 < "0" GOTO PR_CITY
PRINT 00, CO_ADDR16
```



```

PH_CITY PRINT 00, CO_CITY(1), "ICD_STG1", "ICD_ZIP0
PRINT 00,
PRINT 00, "P.O. Number: "IP_0_H00
PRINT 00, : PRINT 00,
PRINT 00, "Please send the following items: "
* IF (ESC) THEN GOTO 3310 TO PRINT 0 ITEMS PER EACH
IF PRINTERS="S" THEN PRINT 00, COMMENT(1) ELSE PRINT 00,
PRINT 00, TAB(2) "Item: TAB(8) "Qty. "TAB(10) "Description" \
TAB(12) "Unit: TAB(17) "Total"
PRINT 00, TAB(3) "0" TAB(11) "Price: TAB(17) "Price"
PRINT 00,
JL=0
FOR JL=1 TO 100
  JLS=JL+1
  IF OTY(JL) = 0 GOTO PRINT1
  IF OTY(JL) < 0 THEN PRINT 00, USING IMAGE5, (ITEM(JL)) : \
  JLS=JL-1 : GOTO NEXT_JL
  ITEM_TOTAL = UNIT_PRICE(JL)*QTY(JL)
  TOTAL = TOTAL+ITEM_TOTAL
  PRINT 00, USING IMAGE(4), \
  JLS, OTY(JL), ITEM(JL), UNIT_PRICE(JL), ITEM_TOTAL
NEXT_JL

```

```

NEXT_JL
PRINT1 IF VAL(SHIPING) <= 0 AND VAL(DISC) <= 0 THEN GOTO PRINT3
PRINT 00, TAB(70) "-----"
PRINT 00, USING IMAGE29, "Subtotal", TOTAL
DISC=(TOTAL*VAL(DISC)/100)+1
IF VAL(DISC) THEN PRINT 00, TAB(70) "-----" : \
  PRINT 00, USING IMAGE29, "Less Discount", DISC : \
  TOTAL=TOTAL-DISC
IF VAL(SHIPING) <= 0 GOTO PRINT3
IF RIGHT(SHIPING,1) = "E" THEN GOSUB CALC_SHIP
ELSE SHIPPING = VAL(SHIPING)
PRINT 00, TAB(70) "-----"
PRINT 00, USING IMAGE29, "Shipping/Handling", SHIPPING
TOTAL = TOTAL+SHIPPING
PRINT3 PRINT 00, TAB(70) "-----"
PRINT 00, USING IMAGE29, "Total", TOTAL
IF PRINTERS = "J" GOTO RTN
PRINT 00,
IF PAY_METH = "V" THEN
  PRINT 00, "Please Charge to a VISA Credit Card 0" : \
  "**** ** ** Exp. Date **/**."
IF PAY_METH = "R" THEN
  PRINT 00, "Please Charge to my Master Card 0" : \
  "**** ** ** Exp. Date **/**."
IF PAY_METH = "C" THEN
  PRINT 00, "Please find enclosed a check for $: \
  PRINT 00, USING "00000.00", TOTAL, "."
IF PAY_METH = "3" THEN PRINT 00, "Please bill us for these items."
IF COMMENTS = "0" OR COMMENTS="0" GOTO PRINT2
PRINT 00, : PRINT 00, COMMENTS : PRINT 00,
PRINT 00, "Thank you."
PRINT 00, : PRINT 00, : PRINT 00, : PRINT 00,
PRINT 00, "Name"
PRINT 00, "Title"
PRINT 00, CHR(10)
RTN TOTAL = 0 : CLOSE 0 : PRINTERS = "J"
RETURN

```

```

CALC_SHIP PERCENT = VAL(LEFT(SHIPING,1)*SHIPPING*111)
SHIPPING = TOTAL*PERCENT/100
RETURN

```

* Subroutine File the Order

```

FILE_ PRINT "The filename will be 'FILE_01'. Is this ok? " : \
00 = INKEY$:PRINT
IF 00 = "N" OR 00 = "n" THEN INPUT "Filename " : FILE_0
OPEN NEW FILE_0 AS 1
PRINT 03, CO_NAME(1), "ICD_ADDR01", "ICD_ADDR01", \
CO_CITY(1), "ICD_STG1", "ICD_ZIP0", "IP_0_H00
PRINT 03, DATE$, "INVT01", "YEAR01", "SHIPPING01", \
PAY_METH$, "ICOMMENTS01", "DISC0
FOR JL=1 TO 100
  PRINT 03, OTY(JL), "ITEM01", "UNIT_PRICE(JL)
  IF OTY(JL) = 0 THEN CLOSE 1 : RETURN
NEXT JL
CLOSE 1 : RETURN

```

* Subroutine Load file

```

LOAD_ INPUT "P.O.# " : IP_0_H00
IF IP_0_H00 = "" THEN STOP
FILE_0 = LEFT$(IP_0_H00,3)+RIGHT$(IP_0_H00,5)
OPEN OLD FILE_0 AS 2
INPUT 02, CO_NAME(1), CO_ADDR01, CO_ADDR01, \
CO_CITY(1), CO_STG, CO_ZIP0, P.O.#
INPUT 02, DATE$, YEAR01, SHIPING01, PAY_METH, COMMENTS01, DISC0
FOR JL=1 TO 100
  INPUT 02, OTY(JL), ITEM(JL), UNIT_PRICE(JL)
  IF OTY(JL) = 0 THEN CLOSE 2 : RETURN
NEXT JL
CLOSE 2 : RETURN

```

* Subroutine Check File for address --Returns W/O 0=0 for not found
0=1 for found
1=0=1=0 then Address is in error.

CHK_ADDR OPEN OLD "ADDRESS.DAT" AS 2

```

INPUT 02, NUMBER_ADDR1
FOR JL=1 TO NUMBER_ADDR1
  INPUT 02, CO_NAME(1), CO_ADDR01(1), CO_ADDR01(1), \
  CO_CITY(1), CO_STG(1), CO_ZIP(1)
  IF LEFT$(CO_NAME(1),3) < LEFT$(CO_NAME(1),3) GOTO CORRECT
  PRINT "Is "CO_NAME(1) the correct company? " : \
  CORRECT0 = INKEY$: PRINT
  IF CORRECT0 = "Y" OR CORRECT0 = "y" THEN OK=0 : "Y" : \
  CO_NAME = CO_NAME(1) : \
  CO_ADDR01 = CO_ADDR01(1) : \
  CO_ADDR01 = CO_ADDR01(1) : \
  CO_CITY = CO_CITY(1) : \
  CO_STG = CO_STG(1) : \
  CO_ZIP0 = CO_ZIP(1) : \
  CLOSE 2 : \
  RETURN

```

```

COR_ NEXT JL
OK=0 : "Y"
CLOSE 2
RETURN

```

* Subroutine Insert new address to address file

INSRT GOSUB BIN_SEARCH

```

NUMBER_ADDR01 = NUMBER_ADDR1+1
IF NUMBER_ADDR01 = LOWER_BOUND01 GOTO INSRT_COUNT
* Move all info from LOWER_BOUND01 to end down one.
FOR JL = NUMBER_ADDR01 TO LOWER_BOUND01 STEP -1
  CO_NAME(JL) = CO_NAME(JL-1)
  CO_ADDR01(JL) = CO_ADDR01(JL-1)
  CO_CITY(JL) = CO_CITY(JL-1)
  CO_STG(JL) = CO_STG(JL-1)
  CO_ZIP(JL) = CO_ZIP(JL-1)
NEXT JL

```

```

* (Insert new address etc.)
INSRT_COUNT CO_NAME(LOWER_BOUND01) = CO_NAME
CO_ADDR01(LOWER_BOUND01) = CO_ADDR01
CO_ADDR01(LOWER_BOUND01) = CO_ADDR01
CO_CITY(LOWER_BOUND01) = CO_CITY
CO_STG(LOWER_BOUND01) = CO_STG
CO_ZIP(LOWER_BOUND01) = CO_ZIP
(replace disk file)
OPEN NEW "ADDRESS.DAT" AS 2
PRINT 02, NUMBER_ADDR1
FOR JL=1 TO NUMBER_ADDR1
  PRINT 02, CO_NAME(1), "ICD_ADDR01", "ICD_ADDR01" : \
  PRINT 02, CO_CITY(1), "ICD_STG1", "ICD_ZIP0" : \
  NEXT JL
CLOSE 2 : RETURN

```

* Subroutine Do binary search on address array --
Returns LOWER_BOUND01 pointing to insert point

```

BIN_SEARCH LOWER_BOUND01 = 1
UPPER_BOUND01 = NUMBER_ADDR1

```

```

BIN_LOOP IF UPPER_BOUND01 < LOWER_BOUND01 THEN RETURN
JL=(UPPER_BOUND01+LOWER_BOUND01)/2
IF CO_NAME > CO_NAME(JL) THEN LOWER_BOUND01 = JL+1 : GOTO BIN_LOOP
UPPER_BOUND01 = JL-1
GOTO BIN_LOOP

```

* Open printer subroutine.

IPM.CH0 is a version of P.CH0 to cause Epson to print in superscript format.)

OPN_PTO PRINT NAME0 : PRINT : PRINT : PRINT : PRINT : PRINT

```

PRINT " (Terminal)"
PRINT " (Serial Printer)"
PRINT " (Parallel Printer)"
PRINT " (Initiator Parallel)"
PRINT " (Bios - SHIP (HEX))"
PRINT " (Output File)"

```

```

PRINT "Which Output Device? " : \
PRINTERS=INKEY$:PRINT
IF PRINTERS="0" THEN OPEN "0.S.CH0" AS 0 : 0=0
IF PRINTERS="P" THEN OPEN "0.P.CH0" AS 0 : 0=0
IF PRINTERS="M" THEN OPEN "0.M.CH0" AS 0 : 0=0
IF PRINTERS="0" THEN OPEN "0.O.CH0" AS 0 : 0=0
IF PRINTERS="0" THEN OPEN NEW "0.PD.OUT" AS 0 : 0=0
IF PRINTERS="Y" THEN PP=12 : HEX Set IF to ASCII Form Feed Char.
RETURN

```

* Error correction routine -- allow re-entry if a character is typed when a
numeric variable has called for.

```

END_H00 IF END<30 THEN ON ENDR GOTO 0
PRINT CHR$(31)CHR$(31)"RETYPE"
RESUME

```

EXT. MEMORY FOR OLD SYS.

Dear Don,

Here is a quickie that I thought might be of interest to the readers. Many of us still have some old memory cards that do not support the extended addressing but we would like to use some more memory for virtual disks (etc.) and really soup up our memory tight systems. And somehow the schematics or layout of all the old memory boards have been lost in the shuffle. Well there is a simple modification one can do to the mother board to decode the extended addresses. It only requires one 74138 chip and a little surgery to the mother board.

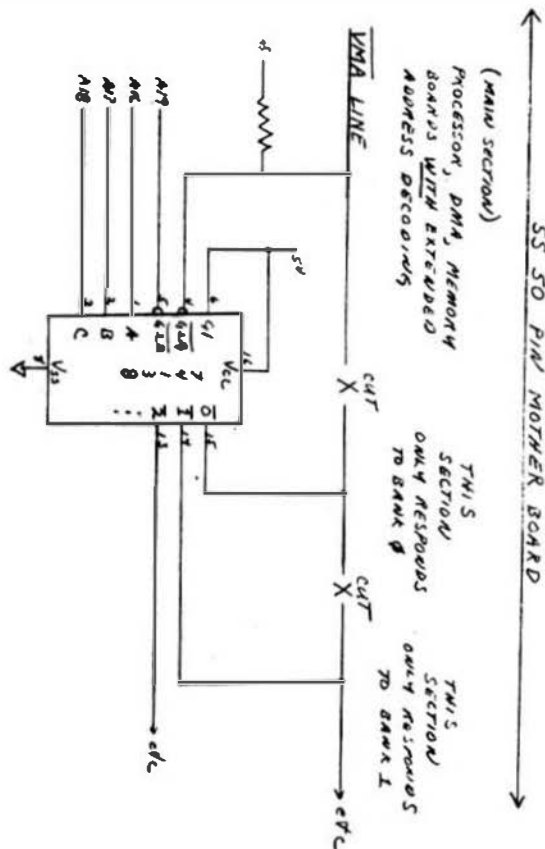
To do the modification you need to decide which slot or slots are to be allocated to which bank(s). The slot(s) will then only respond when their assigned bank is accessed. This modification consists of cutting the VMA* line to segregate those slots from the main buss. The high order addresses along with the VMA* line are then decoded by the 138 as in the figure below to produce the new VMA* line for those slots.

One then can plug any card which does not have extended address decoding on it into one of these slots and it will only respond to the bank which was assigned to that slot.

NOTE:

- 1) The direction of signal flow is determined by which side the processor is on and must match the wiring. (You cannot plug the processor into one of these decoded slots and have the system work.)
- 2) The section of the bus that the processor is on will still function as a normal bus including the extended addressing.
- 3) The I/O section can just as easily be counted as one slot so depending how you define your slots the I/O may or may not get decoded.
- 4) DMA cards and any others which can control the VMA line must be in the same section of the bus as the processor card.
- 5) In most cases there will only be one section (bank 0) since it is not likely that one will have more than 56K of undecoded memory.
- 6) To decode banks from \$0-\$F run A19 to G1 (pin 6) and ground G2B* (pin 5).

EXTENDED MEMORY ADDRESS DECODE ON MOTHER BOARD



Also speaking of virtual memory I hope to have a Virtual Disk Driver for OS9 Level 1 on a SWTPC DAT device for sale by the time this hits the readers (and GIMIX DAT shortly there after.) I expect it to be about \$35 for the source listing and instructions for adapting it to an individual system. Any interested parties can contact me at the address below.

Sincerely,
Matt Scudlere
100 Cedar Ln
Oak Ridge, Tn
37830

MP-C TO MP-T FOR SWPTC

MF-C to MP-T for SWPTC

Hardware modification by

Donnie Wright
5184 Springhill Drive
Pensacola, FL 32503
904 477-8783 CIS 70270,632

One of the most useless things ever produced for the SWTPC 6800 has been around the longest time. You might even have one stuck on a shelf somewhere.

However, back in the "dark ages" of home computing the MP-C control port was quite essential.

For the uninitiated, the MP-C is a "serial" interface that uses a 6820 PIA. Data is clocked out a single bit of the PIA with the parallel to serial conversion being done in software. (aka Milkbug) Data input utilizes a similar scheme.

When the smoke clears after having upgraded an original SWTPC -- with a new cpu and FLEX(tm) installed -- there is this lonely little board lying there...useless! What do you do with it? The problem being that the new monitor doesn't support this software conversion shenanigans. So the old MP-C is just tossed on the shelf and forgotten.

Several modifications for this board have appeared in various publications from time to time. One converted it to a true parallel port. Another replaced the 6820 with a 6850 making it a bona-fide serial board. However, both of these mods involved hardware-intensive changes making the conversion relatively unattractive.

This mod to the MP-C is simple and will make it appear as a MP-T interrupt timer to FLEX's print spooler. Note that it will not completely emulate the MP-T; in the sense that it is not programmable, etc. It will get you print spooling if you are not currently doing so for lack of an interrupt timer. Even if you do not have a MP-C, one shouldn't be hard to find -- and cheap.

The hardware modifications needed for the MP-C consist of merely cutting two traces and installing two additional jumpers. That's it! No additional components are required. Is that simple enough? If so, find your old MP-C, dust it off, and proceed with the following:

Referring to the diagram,

1. Cut the trace running from IC-4 pin 10 to (IC-1) PIA pin 17 (PB7). Make the cut fairly wide and close to pin 17.
2. Jumper from the trace (from IC-4 pin 10) to PIA pin 18 (CB1), moving the connection by one pin.
3. Cut the trace from IC-3 pin 2 to PIA pin 10. This is best done on the top side of the board.
4. On the bottom side, connect a jumper from IC-3 pin 2 to PIA pin 17 (PB7).

Configure the following normal board options:

5. Jumper IRQ to B.
6. Jumper (or select) 110 baud. If this line is no longer 110 baud on your I/O buss, the results will be unpredictable.
7. Jumper the clock lines C1 to C0, either on the board or with a dummy connector plugged in the terminal connector.

Once modified and double-checked, install the board in port 4. Boot the system and make sure you have the files "PRINT.COM", "PRINT.SYS", and "QCHECK.COM" on the system disk. QCHECK is not absolutely essential, but greatly enhances the use of the spooler. Also note that PRINT defaults to the ".OUT" extension. Consult the FLEX manual for documentation on these.

Now, instead of using "P LIST", type "PRINT filename". The printer should take off and you will be returned to FLEX -- free to do other system chores.

If while spooling another program requires disk access, don't be alarmed if the printer stutters or stops for a moment. This is normal as the spooler is a low priority job and FLEX disables it during disk access.

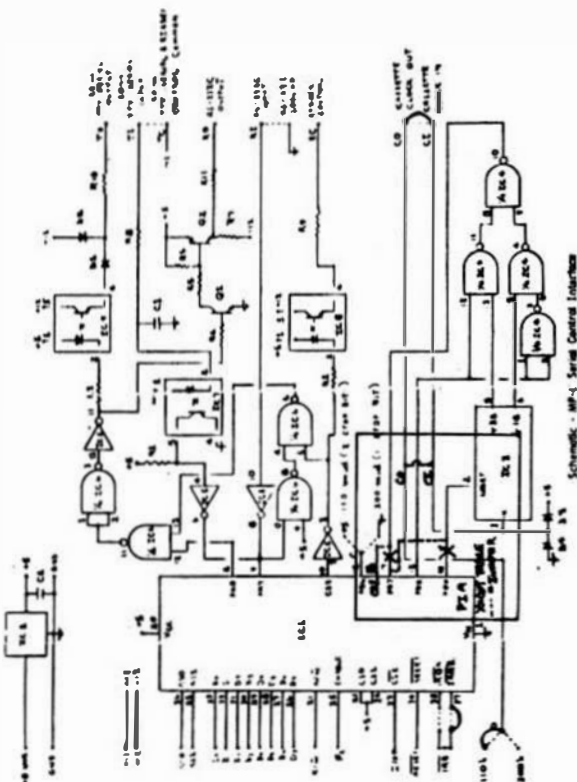
This modification simply uses the baud rate generator to implement interrupts. As configured, the interrupts are approximately 9 msec. This is slightly faster than the 10 msec FLEX configures the MP-T, but the difference is inconsequential. Possibly, even faster baud rates (and interrupts) could be used.

User written programs should be able to make use of the timer without difficulty. Once the PIA is initialized for interrupts, they can easily be toggled on and off by stroking PB7.

Overall, this modification is simple, quick, and puts an otherwise useless board back into worthwhile service. I would be very interested in hearing from anyone who tries it out.

Many thanks to local 68xx guru, Gene Rhodes, for his assistance with the workings of the MP-T and running this piece through his advanced editor.

(tm)Technical System Consultants



WHIMSICAL

Whimsical, a 6809 Compiler

No, I'm not kidding. I just received a package from '68' Micro Journal for a review. It is from a company called Whimsical Developments in Auckland New Zealand. It is a compiler called "WHIMSICAL". The manual is rather unique. It doesn't dive in to the middle of things, but rather gives an interesting introduction to the language. Let me quote a paragraph or two.

"The need for a language like Whimsical for the 6809 was realized soon after the chip became available. It was obvious that the designers had given much thought to software and had done their part by providing us with a machine that had all the stack manipulation, addressing modes, and registers that were needed."

"The possibility of writing all code without significant use of assembly language was potentially a great advantage to the 6809 processor. So Whimsical was born and in keeping with the whole philosophy of the language, the compiler itself could be written in Whimsical."

The introduction goes on to explain how the "bootstrap" process took place, the original compiler being compiled by a version in Algol and later in Pascal. Eventually, enough of the compiler was operative so that it could compile itself, and then be used to add further features.... The discussion is fascinating.

I wondered at first if anyone would or could take a compiler named Whimsical, seriously. (One of the chapters in the manual is titled "Whimsical Statements"). The manual is voluminous, particularly for a "small" compiler. I suppose the suppliers of standard compilers such as Pascal or C can refer their users to Jensen and Wirth, or Kernighan and Ritchie, but suppliers of "something different" have to include a tutorial and introduction to the language as well. The authors of Whimsical have done an excellent job with the manual. The reader gets involved by means of several passes at the language, each one in greater depth.

Whimsical is very much like Pascal, and if you have programmed in either Pascal or C, you will have little trouble using it. First of all, let's get the big question out of the way (at least it is a big question for me). Whimsical does not support floating point arithmetic. It does have a large variety of data types with which many problems may be solved easily and quickly. First there are the unsigned types BYTE and DBYTE. They are just what you might expect from the names. BYTE and DBYTE take Hexadecimal values which are assigned by using the "standard" 6809 symbol '\$': A := \$FF; for example. Next there are three types of integers called SMALLINT (8 bits), INTEGER (16 bits), and LARGEINT (32 bits). Yes, that's right, a full 32 bit integer type. All these are signed, and handle decimal numbers. There are two other types, BOOLEAN, and CHAR. BOOLEAN variables have only two values, TRUE, and FALSE. CHAR types take the values of ASCII characters.

There are conversions to get from one type to another, such as CHR to convert a HEX or decimal number to a character. It is pointed out in the manual that such conversion functions don't really do anything. They are handled at compile time. The point of their use is to keep the programmer thinking about data types (and to catch mismatches in types at compile time also.) You may or may not like rigid typing of variables. I suppose we could get into the standard argument between Pascal programmers and C programmers here. In C you can use a statement such as "putchar(13)", passing what is obviously a decimal number to a procedure that is supposed to handle characters. Pascal (and Whimsical) don't allow such "carelessness" on the part of a programmer. You must use WRITE CHR(13) or WRITE CHR(\$0D) in Whimsical or Pascal.

I have more experience with Pascal than with C, and though it is annoying to have a compiler so stubbornly resist all attempts to "fool it" or take a shortcut, I find that my style of programming gets results sooner with Pascal to catch my dumb errors, than it does with C, which "thinks" that was what I intended to do, and lets me do it.

Whimsical has a few rather neat features. All variables are initialized to zero when the block in which they are declared is "called". That is, the "global" variables, those declared in the beginning of the program, are zeroed when the program is started. Variables declared as "local" within a procedure, are

zeroed each time that procedure is called. All variables must be declared before they are used. Whimsical has the same block structure as Pascal. Procedures may have parameters passed to them by value or by reference. A procedure may be made a Function (to use Pascal's terminology) by putting a variable type in front of its name (the type of the result to be returned). For example, a function to return a random integer might be called: INTEGER RANDOM;. Somewhere in that procedure an assignment statement would have to assign a value to RANDOM to be returned by the function.

The only "strange" difference from Pascal that I noted is that the BEGIN for the main program must be at the top of the program, i.e. the first statement. The last statement must be "END." The period signals end of program. All statements end with a semicolon generally with the rules the same as in Pascal.

There are a couple of rather unique statements in this language. One is called an IF clause. Ordinarily it would be written in Pascal like this: IF A>0 THEN A:=A+1 ELSE A:=A-1;. In Whimsical, the IF clause would make that statement look like this: A:= IF A>0 THEN A+1 ELSE A-1;. This language also supports the rather strange but economical assignment of the result of the evaluation of a logical expression to a BOOLEAN variable. Suppose B is a boolean variable. B := VALUE > LIMIT; is a valid assignment. If VALUE is greater than LIMIT, then B will be assigned TRUE, otherwise B will be assigned FALSE. Of course this is the same as the statement IF VALUE > LIMIT THEN B := TRUE ELSE B := FALSE;. That is in fact the way many languages require you to make that assignment.

Whimsical has the usual mathematical and logical operators, including the MOD operator for INTEGERS, and the AND, OR, and NOT operators for logical expressions. These operators actually work as logical operators when used with BOOLEAN variables, and as bitwise operators when used with BYTE variables. This compiler supports only singly dimensioned arrays, which may be of any of the data types. Comments are preceded by a "%", and the remainder of the line is considered to be a comment. Comments then may be whole lines or may follow statements.

There are all the usual DO WHILE, WHILE DO, FOR NEXT, and CASE constructs available. I/O is via the READ and WRITE statements. These are used to read and write to the terminal, a file, or any other user defined I/O device (defined as a file). If a filename is given in the write statement, output is to that file. No filename or the word OUTPUT or INPUT associate the READ or WRITE with the terminal. It would be hard to envision any simpler and more uniform way of handling I/O. Last but not least, there are several compiler directives to place the Stack at runtime, set the program load address, allow the inclusion of other text files with the source (an include statement), and others to turn the error checking on or off, specify how to handle arithmetic overflow ("roll over" or "saturate"), and several others.

That is most likely enough detail for a review. Perhaps I should add that there are keywords to enable and disable IRQ and FIRQ, and means to set the IRQ RAM vector to point at a procedure in Whimsical. I didn't yet mention that Procedures may be nested on to the extent of allowing sub procedures within procedures. Whimsical supports another "block" level called a module. A module is a group of procedures with all entry points defined. Modules are nestable to any level. The distinction between a module and a procedure is not instantly apparent. The manual indicates among other things, that it is a neat way to keep the variable declarations for the variables associated with a certain program function (not FUNCTION) grouped with the procedures that perform that function.

Running the compiler is a snap. There are a couple of sample programs on the disk that is supplied. One of them is called ASCII. To compile it, you simply type WHIM ASCII. The compiler is large (110 sectors) but once it is loaded the compilation takes place in very short order. If you want to see a listing or output one to a file for printing later, there are options added to the command line. For example for a listing to the terminal you would type WHIM ASCII +L. The compiler is a "one pass" type. It does not require an assembler or a link step, but produces output code directly.

How about a little quick program to see how Whimsical works? I started with the Pascal version of the Sieve Benchmark (again) and I'll include the listing for the Pascal version and the Whimsical version here. My program worked after a couple tries with correction of a couple syntax errors and some nominal misunderstandings

on my part. Now, for the surprise, I ran the benchmark and it times out to just about 9 seconds on a 2 MHz system. That beats everything I've tried up to this point, and beats all the times in the BYTE article (January '83) except the mysterious "IMS Pascal" that I have never heard of or seen advertised.

I am greatly tempted to start my system on a prime program (division algorithm) to find the primes to a very large limit using LARGEINT's, which have a positive limit of 2,147,483,647. That would take a while!

Needless to say, I am impressed with this compiler. I read the manual through once, and I was using it with rather immediate success and minimal problems. As I indicated above, the manual is not "dry" but is easy reading. The introduction is informative and interesting.

With the one reservation that most of my applications require floating point capability, I have to give this one a VERY high rating, realizing that many of the programs that people write, particularly in the area of system programs, may be done very handily without such a capability. It worked as indicated in the manual with no problems whatever. It has LOTS of features that make it possible to get "very close to the machine" without resorting to Assembly code (though there is a way to write procedures in assembler and include them in a Whimsical program too).

With my "bent" toward software for stand alone "ROM programmed" computer applications, I tend to think small, and place emphasis on object code efficiency, simplicity of the compilation procedure, and short compile time. This one passes those requirements easily. The code generated for the benchmark was some 830 odd bytes. Compile time was 35 seconds including the loading of the compiler from my 8" disk. I compiled a longer program and my estimate of compile rate is about 2K of output code per minute. I can only hope fervently that "Release 2 or 3" will include REAL and DOUBLE data types.

Ron Anderson

Editor's Note: Check with South East Media for availability date of delivery of this fine software package.

DMW

% PRIME BENCHMARK PROGRAM IN WHIMSICAL

% SINCE VARIABLES ARE INITIALIZED TO ZERO, WE CAN
% SKIP THE INITIALIZATION OF THE ARRAY BY REVERSING THE
% SENSE OF THE FLAGS

```
BEGIN
  INTEGER SIZE=8190;

  BOOLEAN ARRAY FLAGS [SIZE];
  SMALLINT I;
  INTEGER J, K, COUNT;

  FOR L := 0 TO 9 DO
  BEGIN
    COUNT := 0;
    FOR I := 0 TO SIZE DO
    BEGIN
      IF NOT FLAGS [I] THEN
      BEGIN
        J := I+I+3;
        K := I+J;
        WHILE K <= SIZE DO
        BEGIN
          FLAGS [K] := TRUE;
          K := K+J;
        END;
      END;
    END;
  END;
```



```

        COUNT := COUNT+1;
    END;
END;
WRITE CHR($0D),CHR($0A);
WRITE COUNT,' PRIMES';
END;
END.

PROGRAM PRIME (INPUT,OUTPUT);

( PRIME BENCHMARK PROGRAM IN PASCAL )

CONST
    SIZE = 8190;

VAR
    PRIME : ARRAY [0..SIZE] OF BOOLEAN;
    COUNT,I,J,K,L : INTEGER;

BEGIN
    FOR L := 0 TO 9 DO
    BEGIN
        COUNT := 0;
        FOR I := 0 TO SIZE DO PRIME [I] := TRUE;

        FOR I := 0 TO SIZE DO
        BEGIN
            IF PRIME [I] THEN
            BEGIN
                J:= I + I + 3;
                K := I+J;
                WHILE K <= SIZE DO
                BEGIN
                    PRIME [K]:= FALSE;
                    K := K+J;
                END;
                COUNT := COUNT + 1;
            END;
        END;
        WRITELN;
        WRITELN (COUNT:5,'PRIMES');
    END;
END.

```

TRANSPORT WITH DELIGHT

Transported With Delight.

One of the features I have always liked about Flex is the ease of transporting programs between different systems. So long as the system fully met TSC's specifications for the console and disk driver routines most programs would run without change. The exceptions were usually special test routines that needed direct access to hardware such as the floppy disk controller.

This doesn't seem to be the case any more. I've just spent most of last week modifying a program to get it to run on one of the 6809 systems at work, and I'm not sure its properly fixed yet.

'88' Micro Journal

Its not so long since I had to spend two weeks modifying another program to get it to run. Each program cost around \$200.00, and at this price I would expect the software to run without change. Now that they are running, I'm delighted with them, and can thoroughly recommend them. At the time, I felt like scrapping them.

The problems were undocumented calls to the hardware, and undocumented output to the console. In both cases the software writer assumed the use of a 6850 ACIA for the serial port. This isn't a good assumption, and it limits the market for the program.

In the first place there seems to be a swing away from the use of the 6850 to the newer chips, such as the Western Digital 8250 or the Motorola/Signetics 2661. These both have internal Baud rate generators, and a much wider range of operating modes. They also have a quite different register structure, and need substantially different software drivers.

Secondly, the software can't be used with systems using Interrupt driven I/O to provide such nice features as a type ahead buffer.

Thirdly, the program can't be used with systems where the keyboard is part of the memory map, and is scanned by software.

The systems I work with use an 8250 for console I/O, and a memory mapping scheme that keeps both the monitor and the I/O completely out of the normal address space, leaving the full 64 K of memory free for software. All calls to the monitor, as well as I/O are made through software interrupt calls (SWI).

But back to the problem software. The first program was in two parts, the main and an "Installation" program that lets the customer modify the main program to use the special features of nearly any VDU. (Visual Display Unit or CRT terminal) This is a very attractive approach, as it saves the user from having to directly modify tables or values in the machine code.

Unfortunately, it didn't work in this case. The Installation program ran part way through and then hung up. The distributor was unable to offer a solution, as it was a fairly new product and he had little experience with it. This left the choice of writing to the American company, and waiting for a reply, or dis-assembling the programs and fixing them myself.

It took about two days to dis-assemble the Installation program, learn how it operated, and then patch it so it would run correctly. The rest of the week was spent in dis-assembling the main program and fixing that. In both cases there were direct calls to an ACIA to enable a character to be input from the terminal without echo, and to test if a character had been input.

Its perfectly proper, and in fact it is necessary, to make a direct call in each of these cases, as not all versions of Flex have these routines available. The problem was not that there were direct calls, but that this fact was not documented, and that the routines were not easily accessible so they could be changed to suit the new hardware.

The second program had many more problems. For a start it also used direct calls to the ACIA for both the input and output routines. Although this wasn't covered in the manual, the distributor was aware of the problem, and included details of the necessary patches. Unfortunately, this was a different version, and the routines had been moved.

Strike 1. Not only were the calls undocumented, they weren't even referenced from a fixed location.

The program makes use of various cursor positioning commands, but only allows single character codes for most functions, and two character codes for three functions. The terminal used with this equipment needed an ESCAPE code followed by two or three further codes for each function. Solving this problem took quite a while.

Strike 2. The advertisements for the software don't warn the user of this limitation.

The program was set up and run using another terminal as a temporary measure, and the new terminal immediately went into the graphics mode. There was nothing in the table of values that would account for this, so it was back to the dis-assembler. This time it turned out to be a piece of code that put out a string of characters which appear to be used to set up a SWTP terminal.

Strike 3. There was no mention of this output in the handbook.

The last problem (so far) was much nastier. Just occasionally, the program would crash. This happened over several months. The program was tried on several different systems with the same results. The answer came up almost by accident, when the program tried to access non-existent memory, and crashed. The systems I use have a hardware trap for illegal memory access which stops the program and dumps the registers. The direct page register was set at \$77, and I recognised this as a value set by a program used earlier in the day. Checking the dis-assembled listing again showed that the program used direct page addressing, but DIDN'T initialise the DP register. As a matter of interest, Flex doesn't set the DP register either.

Strike 4. Is it allowed four strikes - or should it have been thrown out before? The programmer must be responsible for initialising the direct page register.

In each case a substantial amount of time was spent in dis-assembly of the software. While I enjoy the intellectual challenge of cracking somebody else's code, and I've learnt a tremendous amount about programming over the years by looking at dis-assembled listings, it shouldn't be necessary.

For a start, consider the costs involved. Assuming a cost of \$25.00 per hour for a programmer, including overhead, the first program cost \$1000 to get running, and the second \$2000. Remember that \$25.00 an hour is significantly less than a software house will charge to do this kind of work, and the real cost is likely to be much higher.

As an engineer I like the 6809 and Flex, as a cost conscious manager I have to make the decision whether to stick with the 6809, or to buy a competing system that can guarantee software compatibility.

Quite apart from the costs involved, there are a number of dangers in having the dis-assembled listings around. Few programmers will be happy to know that their source code has become available to others. More importantly, it becomes very tempting to add those nice enhancements to the software, and to fix those features you don't like. The result is likely to be an orphan copy of the program that isn't properly documented, or worse, one that has input or output that is not compatible with the authorised version.

There are a number of problems that should be fixed before the situation gets completely out of hand. Firstly, there is a proven need for a "get character without echo" routine. If software is going to make use of the power of the newer VDU's, it must be possible to send a string of one or more control characters from the terminal, without these being echoed. TSC has already acknowledged this need, and the adaptable version of Flex has an indirect jump (JMP INCHNE1) through a table at \$03E5.

TSC should consider including INCHNE as one of the normal Flex calls if they haven't done so already, and it could well follow STAT in the table. At the same time, TSC should include a warning in the manual that Flex does not use the direct page register, and that it is the user's responsibility to set this, and possibly, to restore it.

Secondly, and perhaps more importantly, there is an immediate need for a well publicised set of standards or guidelines for writing software for Flex based systems. 68 Micro could be of tremendous help to the industry by drawing up a standard for software that would ensure easy adaption to any 6809 system. Advertisers should be encouraged to say whether each piece of software fully complies with the standard or not. Similarly, reviewers should include comments on how well the software and documentation meets the standard for transportability.

As a starter, I would suggest that the following points be included in the standard.

1. All calls to routines external to the program, or to hardware should be made through the standard Flex jump tables, unless this is completely impossible.

2. Where special routines are needed to provide direct access to the system hardware these shall be clearly identified, and all calls to these routines shall be through a specified jump table.

3. The jump table is to be in a fixed place in the software so its position will not vary with different releases and updates.

4. The preferred position for the jump table is in the first few bytes in the program, immediately after the cold start and warm start addresses.

5. Each entry in the jump table shall reserve four (4) bytes, to allow the use of either direct or indirect jumps.

6. Full specifications are to be given for each of these routines, to allow the final user to write his own versions if necessary. The specifications must include the required action, where the input is to be stored, or the output obtained from, and the registers that may be affected. It must also include the settings for any flags that may be tested during the routine or subsequently.

A following is a typical specification which is based on those given in the TSC Adaptable Flex manual:

INCHNE Put a double byte address pointing to the routine at \$03E5.

This routine should get one ASCII input character from the terminal and return it in the 'A' accumulator with the parity bit (the highest order bit) set to zero. If no character has been entered when the routine is called, it must wait for the character. The

character must not be echoed back to the terminal. Only the 'A' accumulator and the condition codes may be changed.

7. Sufficient memory space must be set aside for the users versions of the routines, and the location and extent of this space must be clearly stated.

8. Any program which uses direct page addressing must set the DP register as required. It would be preferable for the program to save the DP register value on entry, and to restore it before returning to Flex.

If all software followed these rules it would make life much more pleasant. Who knows, I might even have enough money left in the budget to buy more software. I would be only too happy to help in any way I can to get this project moving.

Yours sincerely,

Whitethorn,
3 Lemon Road,
North Balwyn. 3104,
Australia.

Alan M. Fowler.

Editor's Note: This has been approached before. If any of you are "really" interested, please correspond direct with Alan and then let me know what "all" of you decided.

DMW - - -

GSPL

GSPL COMPILER, A REVIEW

Wow! I just received the GSPL compiler for a review. What on earth is GSPL? The manual title is "The GSPL Programming Language by Erick J. Wilner". GSPL is available from Workman and Associates, 112 Marlon Ave. Pasadena, CA 91106. Now to answer the question by quoting the manual introduction.

"GSPL is a systems programming language for medium-large micros. It basically represents a compromise between the power of C and the readability of Pascal, with assorted concessions to the compiler writer. Primitive data types supported are int (signed word), unsigned (unsigned word), short (signed byte), char (unsigned byte), and enumeration types. Arrays and heterogeneous structures are supported as are pointers to all sorts of data objects. The usual control structures are provided (if-then-else, while-do, do-until, case, etc.). There is a fairly complete set of scalar operators, and all type conversions, such as they are, are performed automatically. Little type checking is performed: arrays, structures, and pointers have their types checked when they are used, but there is no prohibition against assigning a pointer a value of a different type, and integers, characters, and enumeration types may be freely mixed. Procedures and functions may be recursive, and may be declared within, and use the local variables of, other procedures and functions."

Rather than a rundown of all the features of GSPL, let me start by describing the manual and the sample programs supplied with GSPL. I happen to have an Epson printer with Grafrax. I have just finished writing a text formatter (in another language) that uses many of the features of the Epson so I am a bit familiar with it and its capabilities. I never, however, thought of designing my own type font using the bit graphics capability of the Epson. Guess what is supplied with the GSPL package? Right, a complete text formatter and a type font file to go with it. The type style is called

ROMAN. It happens that the manual was printed using it. The type face is rather nice, and I had no trouble compiling the source file that is provided and running a test file through the text formatter. There are no instructions or even a list of formatting commands, but the program (in GSPL, of course) uses a CASE statement for the commands and it is fairly obvious which does what.

In addition to the formatter, there is a complete screen oriented editor to allow editing of the type font. I was so intrigued by this possibility that I modified the CRTIO file to match my old ADM, and soon was editing ROMAN and making a new type face I call SQUARE for lack of a better name. It is basically a rather simple straight line character set. It only took an hour or so to get through the alphabet and numbers and I had a new character set to try. I showed the process to my son, and he can hardly wait to try a "script" font.

I was a bit taken aback to see a DO loop terminated with "until Hell freezes over;". Sure enough, half a page earlier I found the definition of a constant "Hell freezes over = false;". Of course this is simply a Repeat Forever loop, since a constant set equal to false, can never be true.

My initial thoughts were that the language is more like "C" than Pascal. I'd have to call it Cascal. The I/O routines for terminal, printer, and files are included in two library files that may be included just as in "C". These are called FLEXIO.1 and UTILS.1. The procedures and functions in these files are described VERY BRIEFLY in the manual. The supplied program examples are a great help in understanding them.

Error messages are both numeric and textual. There is a file called GSPL.ERR that works just like FLEX ERRORS.SYS. It supplies the text for the error message. Some of the messages again reveal some humor. The manual does expand on some of the funnier ones. For example, error #15: Huh? Not very descriptive. The manual adds "You've confused the poor thing. Basically, this means that the compiler encountered a symbol which is not by any stretch of the imagination legal where it is." Error #32 says "You can't do that with those."

In the process of getting the text formatter and font editor running, I of course had to compile the source code since the object was not supplied on the disk. The Formatter has the filename GF (I suppose for Graphics Formatter). Compilation is easy. The command line is simply GSPL GF. Options are available to stop compilation after the compiler has generated Assembler source code, include the source code as comments in the assembler source, turn off the display to the terminal of the name of each procedure or function as it is compiled, and define the number of errors after which the compiler will "give up" and let you fix the source.

The compile operation is totally automatic. The compiler generates assembler code using a scratch file on the working disk. It deletes the scratch file, invokes the TSC assembler (ASMB.COM) on the system drive to assemble the program, and then deletes the Assembler source (provided DELETE.COM is present on your system disk). Therefore you normally end up with the source and the object code only, on your working disk. If you have a problem or want to examine the assembler source, you can stop the compile at that point and assemble it with listing to the terminal or printer.

Though it is easy to invoke the compiler, the compile operation is not ultra fast. On my 1 MHz system it took about 8 minutes to compile the 650 lines of code in GF, (including the library files for I/O and file handling).

One of the library files is called CMDPARSE.1. It provides the facility to get a number of arguments from the command line. Though the manual doesn't say so, CMDPARSE doesn't like commas between arguments, but accepts spaces very nicely. The error message generated by the program gave no clue as to the problem. The solution was discovered by accident.

One of the features of Pascal included in GSPL is the enumerated data type. I'll use the worn out example of "enum days of week (sun, mon, tue, wed, thu, fri, sat) days; days of week is the type name, and days is a variable that can assume the values of the list. What a type definition such as this actually does is to assign the value 0 to the first item in the list, 1 to the second, etc. GSPL has Structures (quite like the RECORD in Pascal) and Unions. A good example of one is given in the FLEXIO.1 file where a FLEX file control block is defined as a structure. A union is like a structure, except that the same area of memory may be used for

several different types of data. To quote the manual "Unions are generally useful for time-sharing memory between multiple objects which don't exist at the same time." GSPL global variables are "static". That is, they exist at fixed memory locations. Local variables and parameters passed to procedures default to "Auto" which means they are allocated on the stack for as long as the procedure that uses them is active. They may be assigned as static, in which case they are treated like the global variables.

Variables may also be declared as "absolute" at an address in memory, a feature that is very nice for accessing I/O ports, etc. GSPL also allows constants to be declared. For use as a system programming language, GSPL's lack of long integer and float types is no handicap at all. My recently completed text formatter, in fact, used only BYTE variables (short in GSPL would be the same). Counting lines and character positions on a page of text requires no more than 8 bits for any counter. (Signed BYTES wouldn't quite do for a 132 column printout).

The author of GSPL decided to permit only lower case for the reserved words in the language. "then" is a reserved word, of course. Upper case is distinct from lower, and "Then" or "THEN" are NOT reserved words. Perhaps the lower case mode is what gave me the first impression that the language is very close to "C". Make it upper case only and it would look a great deal more like Pascal code.

In summary, while GSPL is certainly not for the novice programmer, it is very capable and worthy of consideration as a system programming language. My first impressions are that it is considerably easier to read than pure "C", and that it will be far less frustrating to write programs with it than with "pure Pascal". When I use Pascal, I find myself saying "if I could only..." frequently. Pascal has such rigid type checking that no shortcut is ever possible. I'm quite certain that an experienced "C" programmer would have no trouble with GSPL at all. Experienced programmers who are not familiar with "C" might find the Kernighan and Ritchie book a big help. Workman and Associates could extend the potential market for GSPL considerably if they would prepare a more comprehensive manual that would include a tutorial. In particular, the manual would benefit from the addition of an expanded explanation of the library procedures and functions, and some documentation on the supplied sample programs.

In all honesty, the sample programs are VERY impressive. I had such fun designing my very own character font, that I had great difficulty tearing myself away from the playing to get at writing a program in GSPL to try it out.

The very best way to find out the differences between a familiar compiler and a new one is to translate a program. Since GSPL looks so much like "C", I decided to try it on my favorite Prime Number Program in its "C" version. My favorite test is to find primes to a limit of 10000 (ie range limit, not 10000 primes). This effort brought me a few surprises. I found that GSPL is a bit more like Pascal, and a bit less like "C" than I had originally perceived. First I noted that the main program in GSPL is handled more like it is in Pascal. In "C" it is the last procedure (called main). In GSPL it is considered to be the main program because it is delimited by a simple BEGIN and END, and it is the last part of the program. GSPL uses the words BEGIN and END rather than the curly braces {} and {} that "C" uses to signal the same. "C" uses if (condition) expression, without the keyword "then". GSPL requires the "then" to be present. GSPL also expects the word "do" in a while (condition) do statement. GSPL has a MOD function, the keyword for which is "mod" and not "%" as in "C". Other than those changes, GSPL bears a striking similarity to "C". GSPL's printf function is identical to the one in "C", including a format specification for a numerical or string output. The counted loop structure (usually called FOR-NEXT) is identical to the one on "C". The while-do structure is identical to that of Pascal. Gspl uses do-until for the loop with the test at the end analogous to the REPEAT-UNTIL of Pascal, and adds a new one that eliminates having to use a NOT condition at the end, called a do-while. This is not to be confused with a while-do which has the condition at the beginning of the loop.

After working my way through the syntax errors, I tried compiling my program. In spite of the fact that I couldn't just include utils.l but had to add flexio.l for a few functions such as getch and putch (but I got all the file handling procedures as well), the code generated was about as little as any compiler I have tried on this

program. Just over 3.5K of output code was generated. This is right in there with a couple other recent compiler that I have. Two other compiler that I have generated about 6K for this program. Execution time was very impressive. The last prime was printed just 55 seconds after starting the program. Times for several other native code compilers that I have are 54 seconds, 57 seconds, 58 seconds, and 74 seconds. A P-code compiler that I have, finds primes to 10000 in 194 seconds. Should you want to try the test program, you will need to adjust your results to my system speed. I am running my 6809 at 1 MHz, and my terminal is running at 19.2K baud.

After my program compiled successfully, it ran very slowly. I put a few write statements in it to see what was going on, and found that my flag "save" was not getting set correctly. I thought I had found a bug in GSPL, for several hours and through three versions of my program. It turned out that I had forgotten to change one of the "=" in my "C" program to a ":=" in the GSPL version. As soon as I had that straightened out, the program ran as reported above.

The runtime routines (arithmetic and integer and unsigned comparisons) are supplied in Assembler source code as a library called glib.txt. A look indicates that the arithmetic package is very efficiently written, though I did see a sure simplification in the arithmetic package and a second possible one. (I've reported these to Workman & Associates).

If you'll pardon the pun, all in all this compiler has been written in a workmanlike manner. Operation is very simple. The disk doesn't get cluttered up with numerous files, just the source and the object code. (If you don't specify the -a option, which stops the compiler after the generation of the assembler source file, that file is automatically deleted after it is assembled). The assembler, TSC ASMB.COM on the system drive, is chained automatically. You can simply enter GSPL filename, and go get a cup of coffee. When you return the compiler will be finished without further input from you.

In conclusion, this is a very capable compiler to run in the FLEX environment or in a stand-alone application. Since all I/O is through the library routines, you can easily write your own for other applications. Code is removable or may be made so rather easily. The capabilities are all there. The code was obviously written by an experienced and capable programmer. There were no obvious bugs, no misunderstandings of the 6809 instruction set apparent in the assembler source for the runtime. If you think Pascal is too rigid with its type checking, or feel that "C" listings are a bit too cryptic, you ought to give this one a good hard look.

Ron Anderson

// prime number program (two array version)

@include flexio.i;

@include utils.i;

unsigned number, count;

unsigned saxprime, i, prme[50], prasqur[50];

flag prime, save;

begin

saxprime := 10000;

printf ("%n 1 2 3");

number := 5;

count := 3;

save := true;

prme [1] := 1; prasqur [1] := 1;

prme [2] := 2; prasqur [2] := 4;

prme [3] := 3; prasqur [3] := 9;

while number <= saxprime do

begin

i := 3;

prime := true;

while prasqur [i] <= number and prime do


```

begin
  if number mod prime {i} = 0 then prime := false;
  i+=1;
end;
if prime then
  begin
    printf("%7d",number);
    count +=1;
    if count mod 10 = 0 then printf("\n");
    if save then
      begin
        prime [count] := number;
        primesqr [count] := number * number;
        if count > 26 then save := false;
      end;
    end;
    number += 2;
  end;
  printf ("\nThere were %d primes.\n",count);
end;

end prime2

```

BIT BUCKET

A. MADSEN
THREE-EIGHTY SYSTEMS INC.
P.O. BOX 3088
HARLINGEN, TEXAS. 78550

DEAR DON,

WITH THIS LETTER WE AT THREE-EIGHTY SYSTEMS TAKE PLEASURE IN INTRODUCING OURSELVES TO 68 MICRO JOURNAL AND IT'S READERS WHO SEEM TO SHARE OUR ENTHUSIASM FOR YOUR PUBLICATION AND MOTOROLA PRODUCTS.

IN THE LAST FEW ISSUES OF 68 MICRO JOURNAL WE HAVE NOTED AN INTEREST IN MACROS. WELL, WE HAVE DEVELOPED AN EXTENSIVE MACRO LIBRARY USING TSC'S MNEMONIC ASSEMBLER (THE ONE THAT COMES WITH FLEX), AND SEEING AN INTEREST IN MACRO CAPABILITIES, WE SELECTED ONE OF OURS TO ILLUSTRATE MACRO POTENTIALS WHEN COMBINED WITH CONDITIONAL ASSEMBLY.

WHAT FOLLOWS THIS PARAGRAPH IS AN EXCERPT FROM OUR DOCUMENTATION ON A MACRO WE CALL "TMVC". WE USE THIS MACRO WHEN THERE IS A NEED TO MOVE STRINGS TO DIFFERENT CORE LOCATIONS.

THE TES TMVC MACRO

I. MACRO PHILOSOPHY

OFTEN TIMES A PROGRAM WILL NEED TO MOVE A NUMBER OF BYTES FROM ONE PLACE IN CORE TO ANOTHER. SOMETIMES THE NUMBER TO MOVE IS KNOWN AT ASSEMBLY TIME, OTHER TIMES IT IS

NOT. SOMETIMES SYMBOLIC LABELS ARE AVAILABLE TO DESIGNATE AN OPERAND ADDRESS, OTHERTIMES RELATIVE ADDRESSING IS REQUIRED.

THIS MACRO WILL MOVE BYTES IN CORE FROM THE A OPERAND ADDRESS TO THE B OPERAND ADDRESS FOR WHAT EVER CONDITIONS EXIST AT THAT POINT IN THE PROGRAM.

II. MACRO CALL

TMVC P1,P2,P3,P4,P5,P6

WHERE P1="S" OR "N" IF P1="S" THEN REG B HAS BEEN LOADED WITH THE ZERO RELATIVE MOVE LENGTH. IF P1="N" THEN "N" IS A ZERO RELATIVE NUMBER THAT IS THE MOVE LENGTH.

IF P2="LL" (LABEL,LABEL OPERAND FORM)
P3= A OPERAND LABEL
P4= B OPERAND LABEL
P5= NULL (NOT CODED)
P6= NULL

IF P2="LO" (LABEL,OFFSET OPERAND FORM)
P3= A OPERAND LABEL
P4= B OPERAND OFFSET
P5= B OPERAND BASE REG
P6= NULL

IF P2="OL" (OFFSET,LABEL OPERAND FORM)
P3= A OPERAND OFFSET
P4= A OPERAND BASE REG,
P5= B OPERAND LABEL
P6= NULL

IF P2="OO" (OFFSET,OFFSET OPERAND FORM)
P3= A OPERAND OFFSET
P4= A OPERAND BASE
P5= B OPERAND OFFSET
P6= B OPERAND BASE REG.

III EXAMPLES OF USE

1. TMVC 4,LL,FLDA,FLDB

THIS CALL WILL MOVE 5 BYTES ADDRESSED BY THE SYMBOLIC ADDRESS "FLDA" TO "FLDB"

2. TMVC #63,00,-90,X,0,Y

THIS CALL WILL MOVE 100 BYTES FROM 90 BYTES BEHIND REG X TO 0 BYTES BEYOND REG Y.

3. TMVC S,LO,FLDA,#64,Y

THIS CALL WILL MOVE N+1 BYTES FROM "FLDA" TO 100 BYTES BEYOND REG Y

WHERE M IS A NUMBER IN REG. B

4. TMVC 9,OL,10,Y,FLDA

THIS CALL WILL MOVE 10 BYTES FROM 10
BYTES BEYOND REG Y TO "FLDA".

IV RESTRICTIONS

1. ONLY 16 BIT REGS. (EXCEPT D) MAY BE USED AS BASE REGS. IN THE OFFSET FORM.
2. THE MAXIMUM MOVE LENGTH IS CURRENTLY 256 BYTES.
3. ALL OFFSETS MUST BE WITHIN THE RANGE -32768 TO 32767.
- 4 AS THIS MACRO IS PART OF A SYSTEM, THE USER MUST DEFINE A LABEL THAT IS DEFINED ELSEWHERE IN THE SYSTEM. IT IS A 4 BYTE FIELD LABELED "TESMSP".

V OF INTEREST

1. EXPLICIT MOVE LENGTHS AND OFFSETS (IN THE OFFSET FORM) MAY BE EXPRESSED IN SIGNED DECIMAL, OCTAL, OR HEXIDECIMAL.
2. TESMSP CONTAINS THE 16 BIT ADDRESS OF THE BYTE JUST BEYOND THE LAST MOVED FROM.
- 2, TESMSP+2 CONTAINS THE 16 BIT ADDRESS OF THE BYTE JUST BEYOND THE LAST MOVED TO.
4. ALL REGISTERS ARE PRESERVED ACROSS THIS MACRO.

OK, WITH THAT OVER, IT IS TIME TO QUICKLY GO THROUGH THE MACRO SOURCE ITSELF TO EXPLAIN WHAT IS GOING ON.

LINES 1 THRU 42 DEFINE THE MACRO SOURCE. THE "ASSEMBLER DIRECTIVE" "MACRO" TELLS THE ASSEMBLER TO LOAD ALL SOURCE UNTIL A "ENDM" DIRECTIVE INTO IT'S MACRO TEXT BUFFER BEFORE BEGINNING ASSEMBLY. THEN WHEN DURING THE ASSEMBLY PROCESS, AN OPERATION CODE OF TMVC IS FOUND, SUBSTITUTE THE MACRO SOURCE FOR THAT OPERATION CODE. THIS IS CALLED MACRO EXPANSION. A DESCRIPTION OF THE SOURCE FOLLOWS.

LINES 3 THROUGH 21 ESTABLISH REG X WITH THE A OPERAND ADDRESS, REG Y WITH THE B OPERAND ADDRESS, AND REG B WITH THE MOVE LENGTH. AS FEW AS 2 LINES OF CODE MAY ACTUALLY BE GENERATED HERE (ON P1=S,P2=LL)

OR AS MANY AS B (ON P1=M,P2=00).

LINES 22 THROUGH 25 ASK IF THIS MACRO HAS BEEN CALLED BEFORE IN THIS ASSEMBLY, AND IF SO LINE 23 WILL BE GENERATED TO BRANCH TO THE EXISTING TMVC SUBROUTINE AND LINE 24 WILL HALT THIS MACRO EXPANSION.

IF THIS IS THE FIRST CALL OF THIS MACRO DURING THIS ASSEMBLY, LINE 28 WILL CAUSE THE TEST AT LINE 22 TO BE TRUE FROM THIS POINT ON. IT ALSO WILL GENERATE THE SUBROUTINE (LINES 29 THROUGH 48) THAT WILL BE BRANCHED TO BY SUBSEQUENT MACRO CALLS.

AS CAN BE SEEN, MACROS CAN BE QUITE POWERFUL WITH COMPARATIVELY LITTLE CODE INEFFICIENCY. THEY HAVE THE EFFECT OF RAISING THE LEVEL OF ASSEMBLY LANGUAGE WHILE GREATLY REDUCING LISTING SIZES. WE LIKE THEM. OUR ASSEMBLER LANGUAGE DEVELOPMENT SYSTEM CONTAINS MORE THAN 30 OF THEM AND IS STILL GROWING.

WHICH BRINGS A POINT TO MIND. WHICH EVER ASSEMBLER ONE USES, ONE SHOULD EXPECT THAT HE OR SHE WILL DEVELOP A CLOSE RELATIONSHIP WITH IT. IT WILL HAVE A PERSONALITY THAT IT'S USER WILL HAVE TO BECOME CONVERSANT WITH. TSC'S MNEMONIC ASSEMBLER IS NO DIFFERENT. IF ONE USES THIS ASSEMBLER FOR MORE THAN A FEW MACROS, THEN TABLE ADDRESSES WILL HAVE TO BE ALTERED IN THE PROGRAM ITSELF USING "FIX" OR SOME ALTERNATIVE.

THIS MACRO WAS WRITTEN FOR FLEX AS NOTED EARLIER. WE HAVE JUST GOTTEN UNI-FLEX IN HOUSE AND I FIND THAT THE RELOCATING ASSEMBLER WILL SWALLOW THESE MACROS WITH NO CHANGES. I HAVE NOT ACTUALLY TRIED TO DO THIS BUT THE DOCUMENTATION SAYS THAT IS THE CASE.

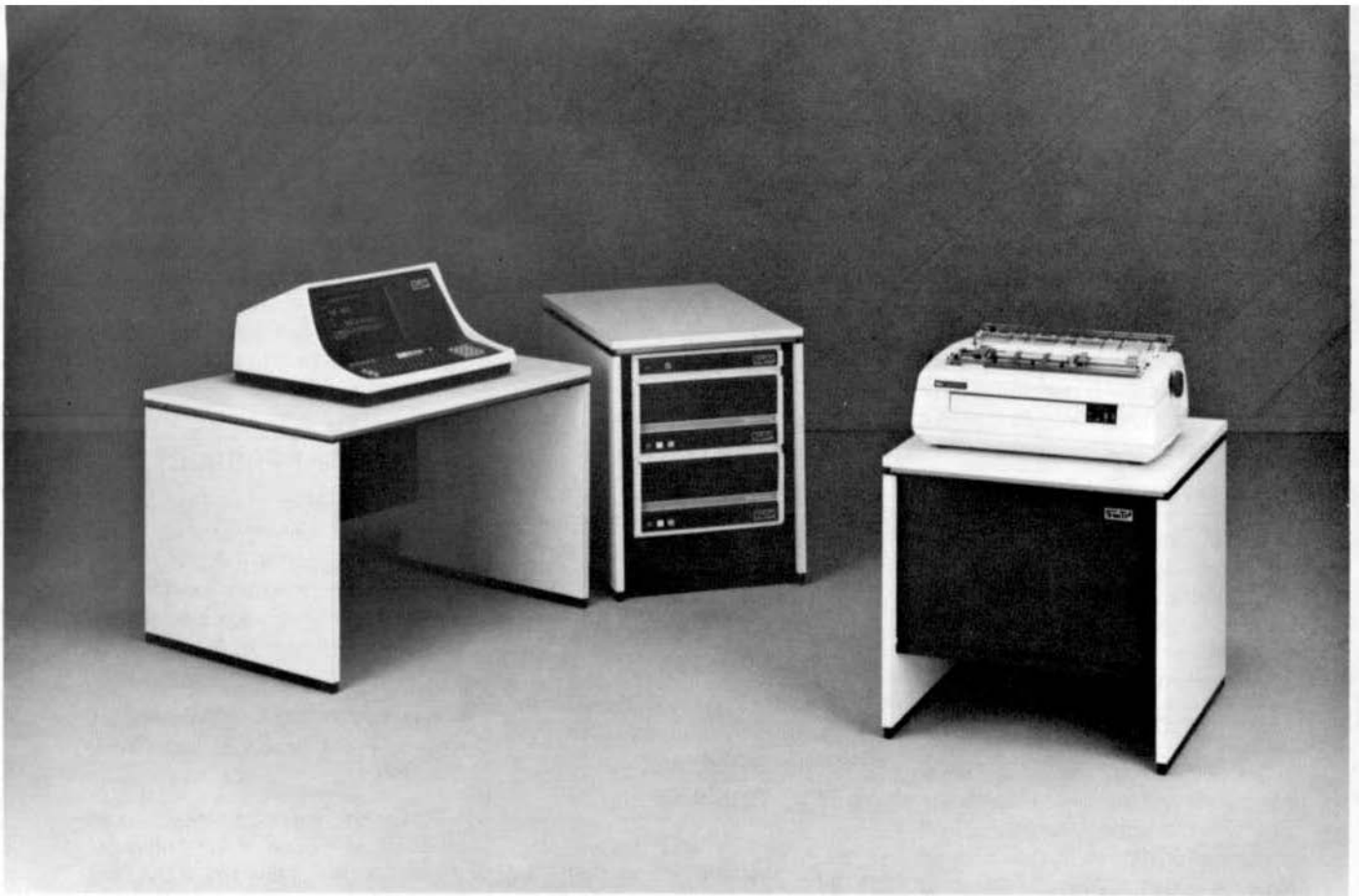
IF THERE IS ANY INTEREST EXPRESSED IN FURTHER EXAMPLES, WE WOULD BE HAPPY TO DELVE FURTHER INTO OUR SYSTEM. WHAT WE HAVE TENDS TO REINFORCE GOOD PROGRAMMING TECHNIQUE WHILE RELIEVING THE PROGRAMMER FROM MUCH TEDIOSNESS.

REGARDS TO ALL

ALAN MADSEN
FOR THREE-EIGHTY SYSTEMS.

OMEGASOFT INDUSTRIAL STRENGTH PASCAL

OMEGASOFT PASCAL	FEATURES	HOST SYSTEM
<p>If you're looking for a language to write real-time process control software, look no further. With the rising cost of labor, it is becoming more critical that a high level language be used. Find out why over 500 companies have switched to OmegaSoft Pascal for their demanding applications.</p>	<p>The compiler generates assembly language for assembly and linking to run on the target system. Since a true relocating assembler and linking loader is used, only those runtime modules required are automatically linked in, providing a much smaller object module than other compilers.</p>	<p>The host system must be 6809 based and have at least 48K of ram (56K recommended) and run one of the following operations systems: MDOS, XDOS, OS-9, or FLEX. Priced from \$425. 68000 host coming in 4th Qtr. 1983.</p>
WHY PASCAL?		SUPPORT PRODUCTS
<p>Pascal was designed to teach students how to write structured programs that are easy to read and maintain. In the past decade it has also proved to provide the same advantages in industrial applications.</p>	<p>Large Pascal programs can be split up into conveniently sized modules to speed the development process. Procedures, functions, and variables can be referenced between Pascal modules and assembly language modules by using Pascal directives.</p>	<p>The OmegaSoft Relocatable Assembler and Linking Loader is designed to support the Pascal Compiler Package and can also be used for general assembly language program development. Priced from \$125.</p>
EXTENSIONS	Full source code is included for the runtime library, the debugger, and other support utilities.	<p>OmegaSoft's Screen Editor supports smart terminals and comes complete with the Pascal source. Priced from \$90.</p>
<p>OmegaSoft has taken the Pascal framework and expanded the basic data types, operators, functions, and memory allocation to fit the needs of real-time systems. These additions fit in the same structure as Pascal and enhance its usefulness without impairing the excellent readability, ease of maintenance, and structured design.</p>	ISO COMPATIBILITY	<p>Faster floating point execution can be obtained by use of the Arithmetic Processor Unit option which uses the AMD9511 APU chip. Priced from \$90.</p>
<p>The byte data type allows you to directly address bytes in memory or I/O devices. The common arithmetic operations can be used for bytes along with shift left, shift right, "and", "or", "eor", and complement operators. These operators are also available for integer and hex (2 byte unsigned) numbers.</p>	<p>OmegaSoft Pascal has been tested using the pascal Validation Suite. The Suite is a collection of over 400 Pascal programs designed to test the quality of Pascal Compilers and their runtime systems for compliance with the ISO (International Standards Organization) Pascal standard. OmegaSoft is the only supplier of 6809 native Pascal compilers that publishes this report in its instruction manual.</p>	<p>For complex real-time applications, the Multi-tasking Kernel provides task scheduling, inter-task communications, and resource interlocking. The Kernel is a runtime library that is accessible as Pascal functions and procedures (with full source included). Priced from \$175.</p>
<p>LongIntegers are four byte signed numbers useful for extended range arithmetic commonly needed for machine control. Functions have been added to allow conversion between the various data types. Dynamic length strings allow complex text manipulation and allow effective interactive I/O.</p>	DEBUGGER	<p>Dealer and OEM inquiries invited. OmegaSoft products are also available from distributors in Australia and Western Europe, call or write for more information.</p>
<p>Variables can be placed either on the data stack (default), at an absolute address in memory (for I/O), in base page, relative to the program counter (for constant tables), or defined in another module.</p>	<p>The compiler package includes an interactive, symbolic debugger. The debugger allows setting of breakpoints, displaying and changing variables, and tracing statements. The debugger allows very fast turnaround for programs to be run on the host system.</p>	<p>OMEGASOFT P.O. Box 842 Camarillo, CA 93010 (805) 987-6426</p>
	TARGET SYSTEMS	<p>TM: MDOS and XDOS are Motorola trademarks. FLEX is a trademark of TSC. OS-9 is a trademark of Microware.</p>
	<p>The target system may be any 6809 system. No specific I/O devices are required. The output code is re-entrant and rom-able, perfect for single-board systems up to large development systems. There are no charges for use of the output of the compiler or the object of the runtime library in your products. 68000 target coming in 3rd Qtr. 1983.</p>	



THE COMPLETE BUSINESS SYSTEM

+ Multiuser + Highly Expandable + Cost Effective

S+ THE CONCEPT

The S+ system is a modular computer system in which all portions of the hardware and software are designed to work together in the most efficient way possible. An S+ single user system with floppy disk storage is a competitive and cost effective entry level system. Unlike most other small computers being sold as "personal", or "small business" machines, the S+ system may be expanded to maximum capabilities using this same hardware and software. You cannot end up with a DEAD END system that cannot be expanded and whose software is not compatible with larger machines. A basic S+ system may be expanded to thirty-two users, a megabyte of main memory and hundreds of megabytes of hard disk storage by simply plugging in, or connecting the desired upgrade equipment.

TOTAL DESIGN—Hardware and Software

The S+ system is an integrated hardware and software design. The two complement and enhance each other in this system. The UniFLEX® operating

system used in the S+ systems is patterned after the Bell Laboratories UNIX® operating system, one of the most admired and widely used operating systems in the world. Instead of being an afterthought, the software is part of the design of the S+ system. You can be sure that with this approach that all parts of the computer operate with maximum efficiency and cost effectiveness.

THE CENTRAL PROCESSOR

The basic S+ system is configured with 256K bytes of memory and can be expanded to more than 1 million bytes. An efficient and fast hardware memory management system is used to allocate the available memory among the users on a dynamic basis. As little as 8K bytes, or the entire memory—if needed—can be used by any individual user. This makes it possible to run very large programs on the system, but it also uses no more memory than necessary for a particular job. The increase in cost effectiveness of this system over crude and outdated bank switching arrangements is dramatic.

The central processor runs in both user and supervisor states. It can detect and reject a defective user program. It is impossible for a user program to go bad and stop the entire system, as can happen quite easily in less sophisticated systems.

Task switching is accomplished by use of a multiple map RAM memory, with sixty-four individual task maps. Each task can access from 4 to 64 K-bytes of memory. Multiple tasks may be used in programs that require more than 64K bytes of memory for execution. When a task is completed the memory is automatically released for other use.

SOFTWARE

The S+ operating system, UniFLEX® is a multiuser, multitasking operating system based on the UNIX® operating system that has been used for many years on Digital Equipment Corp. PDP-11 series minicomputers. It is considered one of the most sophisticated and "user friendly" operating systems available. Variations of UNIX® are rapidly becoming standard on mini and larger microcomputers.

A large variety of languages are available for use with the system. These include FORTRAN, COBOL, BASIC, and Pascal. Word processing packages are also available to give you full text processing capability on the system.

Applications programs are available in large quantities in many fields. This includes general business, medical, dental, veterinary, library and real estate management; plus others. Since the system is multiuser it can also be connected to cash registers to produce a point-of-sale terminal system combined with the computer. The possibilities for application of this system are endless.

THE I/O SYSTEM

The S+ system is totally interrupt driven. All terminal and printer I/O devices connect to an I/O bus separate from the main bus. Up to thirty-two separate devices may be connected to the I/O bus at any one time. If I/O activity is great enough to cause an unacceptable slowdown in system operation, a separate I/O processor can be installed in the system. This plug-in option removes all I/O handling

overhead from the main processor and allows operation of up to thirty-two external devices at 9,600 baud. Without an integrated total design, as in the S+ system, it would become impractical to use a UNIX® type operating system in a situation with heavy terminal I/O activity.

DISK STORAGE

A wide range of disk storage capacity is available for the S+ system, from 2.5 M-byte floppy disks to an 80 M-byte Winchester and many sizes between. All disk controllers use direct memory access (DMA) type operations to maximize data transfer and to minimize overhead on the main processor. The Winchester disks also use intelligent controllers along with DMA transfers to preserve the performance that these type devices are capable of giving. Without this distributed intelligence the system performance would be greatly degraded. The UniFLEX® operating system is designed to work at maximum efficiency with this type disk system. The data transfer rates achieved by this combination rival those of large minicomputers.

COMMUNICATIONS

A high speed local network communications system is available to interconnect S+ systems. The VIA-BUS® network will allow communication between systems at data rates of over 400K baud. Such a system makes it possible to share data between local systems in an efficient and low-cost manner.

AVAILABLE SOON

Tape backup—20M-Byte in less than 15 minutes on a standard ¼ inch cartridge.

Mini-Wini—5 and 10 M-Byte Winchesters—5¼ inch package. Winchester performance, for smaller systems in a small package. UniFLEX® compatible design.

Large Capacity—190 and 340 M-Byte Winchesters, plus SMD cartridge drives.

UniFLEX is a registered trademark of Technical Systems Consultants, Inc.

UNIX is a registered trademark of Bell Labs.

VIA-BUS is a registered trademark of Southwest Technical Products Corporation.



SOUTHWEST TECHNICAL PRODUCTS CORPORATION
219 W. RHAPSODY
SAN ANTONIO, TEXAS 78216

(512) 344-0241

```

TMVC MACRO
PSMU X,Y,D SAVE USERS REGS
IFC &1,S,1 IF LENG. SET IN B REG., SKIP 1
LDB &1 * LOAD PARM LENG
IFNC &2,LL,2 FOR "LL" LABEL, LABEL FORMAT
LDX &3 * LOAD "A" OPERAND ADDR
LDY &4 * LOAD "B" OPERAND ADDR
IFNC &2,LO,2 FOR "LO" LABEL, OFFSET FORMAT
LEAY &4,&5 * LOAD "B" OPER. (OFFSET FORM)
LDX &3 * LOAD "A" OPER. (LABEL FORM)
IFNC &2,OL,2 FOR "OL" OFFSET, LABEL FORMAT
LEAY &3,&4 * LOAD "A" OPER. (OFFSET FORM)
LDY &5 * LOAD "B" OPER. (LABEL FORM)
IFNC &2,OO,7 FOR "OO" OFFSET, OFFSET FORMAT
STU TESMSP * SAVE U
LEAU &3,&4 * GET "A" OPER. ADDR (OFFSET FORM)
STU TESMSP+2 * * SAVE IT
LEAU &5,&6 * GET "B" OPER. ADDR (OFFSET FORM)
EXG Y,U * * LOAD "B" OPER. ADDR
LDU TESMSP * RESTORE U
LDX TESMSP+2 * LOAD "A" OPER. ADDR.
IF TESM1=1 IF TMVC PREV. CALLED
LBSR TESOL * USE IT'S SUBROUTINE
EXITM * AND END MACRO COMPILE
ENDIF
BSR TESOL EXECUTE MACRO SUBROUTINE
BRA TESOM * TO MSI
TESM1 SET I SET MACRO CALLED
TESOL EQU * TMVC SUBROUTINE
LDA ,X+ * GET BYTE TO A
STA ,Y+ * MOVE TO REQUIRED ADDR
TSTB * TEST AND BRANCH IF DONE
BEQ TESOM * * ON 0 EXIT
DECB * REDUCE MOVE COUNT
BRA TESOL * LOOP UP FOR NEXT
TESOM EQU * EXIT MACRO
STX TESMSP * SAVE LAST ADDRESSES
STY TESMSP+2 * *
PULU X,Y,D * RESTORE REGS
RTS * RETURN TO USER
TESOM EQU *
ENDM
548 *
0177 36 36 TMVC 10.00,-50.Y,10.X
0177 36 0A PSMU X,Y,D
0179 C6 0A LDB #10
017B DF AB STU TESMSP
017C 33 AB LEAU -50.Y
0180 DF AD STU TESMSP+2
0182 33 0A LEAU 10.X
0184 1E 23 EXG Y,U
0186 DE AB LDU TESMSP
0188 9E AD LDX TESMSP+2
018A 8D 02 BSR TESOL
018C 20 12 BRA TESOM
018E A6 80 0001 TESM1
0190 A7 A0 018E TESOL
0192 5D EQU *
0193 27 03 LDA ,X+
0195 5A STA ,Y+
0196 20 F6 TSTB
019B 9F AB BEQ TESOM
019A 109F AD DECB
019D 37 36 BRA TESOL
019F 39 EQU *
019B 9F AB EQU *
019A 109F AD EQU *
019D 37 36 EQU *
019F 39 EQU *
01A0 TESOM EQU *
ENDM

```

```

01A0
01A0 36 36 TMVC S.LL,MOD2,MOD3
01A2 8E 01CA PSMU X,Y,D
01A5 10BE 01D4 LDX #MOD3
>01A9 17 FFE2 LDY #MOD3
LBSR TESOL
ENDM
551
552 *
01AC 36 36 TMVC #63,00,1000,Y,0,X
01AC 36 36 PSMU X,Y,D
01AE C6 63 LDB #63
01B0 DF AB STU TESMSP
01B2 33 A9 03EB LEAU 1000,Y
01B6 DF AD STU TESMSP+2
01B8 33 B4 LEAU 0,X
01BA 1E 23 EXG Y,U
01BC 1E AB LDU TESMSP
01BE 9E AD LDX TESMSP+2
>01C0 17 FFCB LBSR TESOL
ENDM
553
554 *
555 *
0177 *
01A0 *
01AC *
PAG
TMVC 10.00,-50.Y,10.X
TMVC S.LL,MOD2,MOD3
TMVC #63,00,1000,Y,0,X
PAG

```

SINGLE BOARD 6809 COMPUTER

Sanasaka Systems
3311 Concord Blvd.
Concord, CA 94519
June 1983

6809 - A 6809 COMPUTER ON A BOARD

This article describes the design and construction of a 6809-based single board micro-computer. I designed this computer to replace my old one, which I was afraid would fail at any moment (it did fail during the final debugging of the one described here). My original computer was a home brewed kludge using the \$100 bus and an MEK6800-D2 evaluation kit. The computer used memory mapped video and a parallel connected keyboard with dual cassette tapes for storage. This was later modified to support a 6809 and FLEX9 (FLEX is a trade mark of Technical Systems Consultants) with 2 5-1/4 discs.

GENERAL DESIGN CONSIDERATIONS

Before getting into the details of the design I'll give you a little of the philosophy behind the design. My original computer was bus based to allow room for the addition of all sorts of goodies later. As time went on, the only thing that I added were the discs and 8K of RAM. The rest of the system remained as originally designed and built.

In the spring of 82 I decided the time had come to think about a replacement. After looking at the prices on a system incorporating the functions I wanted, I decided I couldn't afford to buy one. I basically wanted the full 65K address space, discs, memory mapped video, FLEX compatibility and several I/O ports. A system providing this is over \$2000; thus the decision to design and build my own.

Being a great believer in not reinventing the wheel I hunted for a circuit I could either copy or modify easily. I found one in the Motorola application note AN-851 "MOTOROLA MC6845 CRT/ SIMPLIFIED VIDEO DISPLAY CONTROLLERS". The application note details the design of a video terminal using the MC6845 and includes an almost complete schematic for a CRT terminal, which also happens to be a special purpose computer. After some study of the circuit and another application note (AN-830 "AN INTELLIGENT TERMINAL WITH DATA LINK CAPABILITY") I decided that the circuit in AN-851 could be expanded to provide the functions I wanted.

My requirements were:

1. 6809 CPU
2. FLEX compatibility
3. Memory mapped video (to use my monitor and avoid having to buy a terminal)
4. Keyboard and printer parallel ports
5. One or two additional parallel ports for other functions
6. Serial port (RS-232)
7. Disc interface to use my existing 8230 controller and discs
8. 7K EPROM space (for my own 7K monitor program, SANBUG09)
9. Maximum amount of RAM (65K-7K EPROM -1K I/O space -2K video refresh space - 56K)

The computer would be built on a breadboard so that it could be expanded by adding circuits in the same way as the original circuits.

In order to arrive at the functions described above, several changes had to be made to the circuit of AN-851. I changed from a 6808 to a 6809, added more memory (RAM and EPROM) and added I/O addressing. The video circuitry, about which I know nothing, was left unchanged. (More on this later.) CRT screen refresh memory area was reduced from 8K to 2K in order to not occupy more of the program area than necessary and still provide an 80x24 display.

DETAILED DESIGN

The design was fairly straightforward from the application note schematic. I decided to use the address range of the 6530 bus for my I/O, to remain compatible with most of the rest of the world. I set up the decoding of the first 8 addresses to be 4 bytes apart because this gave the maximum efficiency to the basic functions to be included and would be compatible with the FLEX9 disk drivers I had been using on my old system. Additional decoding could be supplied at greater aspiration later if desired. On this basis, I set up the memory map almost as in Fig. 1. The RAM shown above \$8000 was added after I decided to use dynamic RAMs (DRAMs). The memory mapped video was put at \$8000 to be below FLEX and still be as much out of the way as possible. It was not put above \$8000 because I wanted the entire area between \$4000 and \$7FFF available for SANBUG09.

The biggest problem to be resolved was the memory. The EPROM was no problem. I used 7327's because they were readily available and cheap. Two of them took care of the 7K of EPROM with only 1K thrown away. RAM was

another matter however. The simplest was the 6116 static CMOS RAMs, but they were around \$10.00 each, a total of \$730.00 for 64K. In addition, the 24-pin packs take a lot of space and wiring to connect. The alternate was 65K dynamic RAMS (DRAMs). They were (spring 82) about \$9.50 each (\$76.00 total), and didn't take much room BUT... DRAMs were hard to make work, and require lots of support circuits, or so I'd heard.

The price difference between the static and dynamic RAMs was too great to ignore. Some preliminary investigation showed that 64K DRAMs are not as hard to use as I'd thought. Since I was going to be getting an unknown brand of DRAMs I designed for the worst case, i.e., no internal circuits for automatic or hidden refresh or connecting input and output. Careful study of the timing for a 1Mhz 6809 in relation to the timing for 250ns 4164 DRAMs showed no problems in using them together.

To determine how to implement the DRAM refresh and multiplexed access required study of the timing for both the 6809 and DRAM. The result was Fig. 2. Memory Timing. Study of Fig. 2 reveals how the memory timing had to work. The top two lines are E and O, respectively, right from the 6809 boot. The read data and write data lines indicate the requirements to keep the CPU happy for reading and writing.

The design data for the 4164s shows that they need to be refreshed every 1 to 4 milliseconds, depending on brand, but could be refreshed more often if desired. Data is written into the DRAM by the falling edge of CAS when the write signal is already low, as is the case with the '09. Data is valid from the DRAM when the write signal is high, following the falling edge of CAS and while CAS remains low. Refresh is accomplished by strobing each of the 256 rows with RAS at least once each refresh period. The multiplexing of the refresh addresses with the data access addresses is currently handled, in 16K DRAM chip designs, by a DRAM controller chip. I was not able to find one I could afford that would handle 65K DRAMs so I designed my own address multiplexer and refresh counter.

Fig. 2 shows that the time to refresh is when both E and O are low. At this time address, control and data signals are not valid and the CPU is not accessing memory. Data write and read is somewhat more complicated.

Write data is valid on the rising edge of E, and data is read on the falling edge of E. From this, it appears that the rising edge of E is the place for CAS to fall, and write the data into the DRAM. In order for this to work RAS, must already have been low for at least 35 ns. Therefore, if RAS were to go low on the rising edge of E, and CAS to go low at least 70 ns (35ns hold after RAS - 35ns setup before CAS) later, the write of data would occur 70+ ns after E goes high. Since, during a read cycle, data must be valid until at least 10 ns after E goes low, CAS must also be held low for at least that long.

Using the above discussion as a basis, the last two lines, RAS and CAS, were added to Fig. 2, and it became apparent how they can best be generated. RAS must be low during refresh and again before CAS goes low. After CAS is low, RAS can go high. With this it appears that RAS can be low when both E and O are low, refresh, and low again when E and O are both high (RAS = E .EOR. O). Since CAS low is only required for data access, and I previously showed it must lag RAS low by at least 35ns and E low by 10ns, it appears that if it lags E by 70ns and is of opposite state, all access timing should be correct.

What about refresh? Since the refresh addresses were to be gated to the chips every CPU cycle, why not use the CPU clock to run the refresh counter? That's exactly what I did. The schematic, Fig. 3, shows the counter (IC37 & IC38) is incremented by E during the time when the CPU is accessing the memory. The DRAMs are refreshed more frequently than necessary, but it keeps things simple.

The DRAM chips have their addresses multiplexed so it is necessary to gate the correct 1/2 of the address bus to the chips as well as the refresh address. Again looking at Fig. 2 we can see that refresh must be gated during E and O low, when O goes high AO-A7 must be gated to be ready when RAS goes low. Next, A-A15 must be gated 35ns after RAS goes low to be ready when CAS goes low. After CAS goes low the addresses don't matter until ready for refresh. Because AO-A7 must remain valid for 35ns after RAS goes low E can't be used to switch to A8-A15. By using an E that's delayed from the CPU E the switch can occur after the 35ns and still be ready for CAS going low.

Fig. 3 shows the gating used to handle the refresh and data addresses for the DRAMs (IC33-36). The delay for E [E delayed] is generated by the propagation delay in 2 7404 inverters and the 74133 1 of 4 multiplexer, while the delay for CAS is a result of 3 additional 7404 propagation times beyond ED.

The final point in the memory design was the use of IC16 and 17 (8276's). These were used because when I was doing the design I was not sure if the DRAM chips I would be getting would allow tying input and output together. See also Motorola AN-835 "64K DYNAMIC RAM MEMORY BOARD WITH TRANSPARANT ADDRESS".

The DRAM memory design was tested by running a memory diagnostic designed for dynamic RAMs for over six hours without any errors.

After designing the memory using 64K DRAMs, I realized several additional advantages. If I enabled DRAM every time the CPU did a write to ANY location, the data could be read from the DRAM rather than whatever was actually written to. This was particularly useful for the control port, which is write only. In addition, by reading and writing the EPROM addresses, the program in the EPROM could be copied into the DRAM at the same addresses. Then, if the EPROM were "turned off", it would allow easy modification or replacement of the EPROM program. In addition, the area between \$E100 and \$E3FF was made available for use by the SAMBUG09 thereby satisfying its RAM needs (it only uses \$E300 through \$E3FF).

The computer was designed for SAM UG09 to be resident in the top 7K of the address space. The ability to turn off the EPROM under program control, and use the underlying DRAM opens an additional area for special programs to replace SAMBUG09 and not impinge on the normal program run area.

The computer also includes a computer control port which, as mentioned above, is write only. It is also only 4 bits wide. The bits are D0 - EPROM on/off, D1 - SW1 counter on/off, D2 - bell and D3 - spere. This port is used to avoid tying up any of the bits on the 2 FIA's. The SW1 counter is used by the monitor program in its single step routine and in setting breakpoints when exiting to a user program.

The memory mapped video is supported by a 2K 6116 CMOS RAM addressed between \$B800 and \$BFFF. A separate RAM is provided to avoid contention problems between the CRTIC and the CPU during times the CPU is not addressing the screen. By writing to both the video refresh RAM and the DRAM, any reads of data from the CRT would not cause problems with "glitching" on the screen because the read data would come from the DRAM not the refresh RAM. Writes to the screen are synchronized with horizontal retrace by means of the SYNC instruction.

I mentioned above that I did not change the video circuits because I knew nothing about them. After I got the prototype running, using the ACIA, I proceeded to determine the numbers to use to set up the CRTIC (4843). After such experimenting I found that if I wanted 24 character lines, each line could contain only 11 horizontal traces. This was because the horizontal oscillator in my monitor is not fast enough (only 15750KHz) to allow 24

lines of 17 traces each. The circuit in the application note was designed for 12. When 17 traces per line were used for 24 lines the screen refresh rate was less than 60 times per second and the image wavered unacceptably.

The character generator chip I was using IC720 originally a 66714; needed 12 traces per line to properly form characters with descenders. When only 11 lines were available the bottom part of several characters were cut off. I tried changing to a 66720 chip, which only requires 9 traces, but found that I didn't like the appearance of some of the special characters or the characters with descenders. As a result, I decided to replace IC70 with a 2732 using my own character set designed for 11 traces per line. This had the additional advantage that I could now create 128 characters beyond the 128 character ASCII set.

The character generator generates characters on the CRT screen as follows:

1. The CRTIC selects the cell on the screen to be refreshed and places that address on its address bus (MAC-MA10).
2. The CRTIC also places the row address on the row select lines (AMB-AS4) with row 0 being the top row of the character.
3. The refresh address selects the refresh data in the refresh RAM.
4. The refresh data (100-1FF) selects the character to be displayed by providing the upper 8 bits of the character generator address (D0 is A4 ... E7 is A11).
5. The row address from the CRTIC provides the lower 4 bits of the character generator address, which defines the row of the character to be displayed (RA0 is A0 ... RA3 is A1).
6. The data for the row of the character addressed goes from the character generator to a parallel to serial converter (IC7) from which it is clocked to the screen.

From this description you can see that each character on the character generator occupies 16 address locations (4 row address lines define 16 addresses). Each EPROM location defines the dots on one row of a character. Since the character height is only 11 rows 5 rows of data are not used. As an example lets see how the letter A is generated at the top left of the CRT screen using a 2732 EPROM as the character generator.

The CRTIC sends out the refresh address of \$0000 which causes the refresh RAM to place the ASCII code for the letter A (\$40, on its data output. This in turn addresses location \$40X in the character EPROM. The CRTIC also puts out a row address of 30 which is combined with the \$40X above to yield a final EPROM address of \$400. The byte at \$400 in the character EPROM is the dot pattern for the top row of the letter A (\$04). After the CRTIC has scanned 80 refresh RAM locations for the top row of data for the top line on the screen it repeats the same 80 refresh addresses for the next 10 rows of data. Thus when the refresh address is again \$0000 but the row address is \$1 the byte at \$401 in the character EPROM is addressed. This byte is the dot pattern for the second row of the letter A (\$14). Since the characters are each only 7 columns wide only 7 of the bits in each byte are used.

Thus the organization of the EPROM has to be 256 blocks of 16 bytes each (4K), with each block representing a displayable character for the CRT screen. The first byte (0) of each block is the top of the character to be displayed and the 15th byte (10) is the bottom of the character. Bytes 11 through 16 are not used. To build another character set all that needs to be done is set up the data for another EPROM.

OPERATION

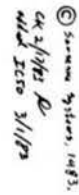
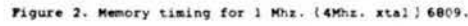
Operation is simple. After the power is turned on, reset is released by an RC timer. SAMBUG09 begins by initializing itself. The program assumes that the CRTIC is present, that a keyboard is connected to FIA1A, that horizontal sync is connected to CAJ, and vertical sync is connected to CA2 of the PIA at address \$E004. After the CRTIC is initialized, the horizontal and vertical pulses should be available on the PIA status port. This is tested, and if the pulses are present, the keyboard and CRTIC are set up as the console device. If the pulses are not present, then the program assumes that the ACIA at \$E000 is the console device, and it is set up accordingly. After initialization, the monitor loop on and prompts for user input. Typing sec-V causes FLEX09 to boot in from a 5 1/4 inch disc.

I have described the development of a very powerful computer for very little money and a LOT of work. It is definitely not a job for a beginner. When the construction was completed it required an additional 10 to 15 hours and some sophisticated test equipment to get all the bugs out. Since some of the bugs were design related they do not appear in the schematic and if you build the computer your trouble shooting should be only that required to find construction errors.

If you would like a full size copy of the schematic and a complete parts list, send \$10.00 to Sensake Systems, 3311 Concord Blvd., Concord, CA 94519. SAMBUG09, the character EPROM and software support are also available.

ADDRESSES	FUNCTION
0000-1FFF	User DRAM
B800-BFFF	Memory mapped video
C000-FFFF	DRAM (FLEX09)
DF00-FFFF	SAMBUG09 pointers
E000	ACIA status/control
E001	ACIA data I/O
E004	FIA1A data
E005	FIA1A control
E006	FIA1B data
E007	FIA1B control
E008	FIA2A data
E009	FIA2A control
E00A	FIA2B data
E00B	FIA2B control
E00C	FIA1A data (future)
E00D	FIA2A control (future)
E00E	FIA1B data (future)
E00F	FIA1B control (future)
E010	Computer control port
E014-E017	SE30 board sel (diac)
E018-E01B	SE30 board sel (diac)
E01C	CRTIC register select
E01D	CRTIC registers
E020-7F	Unassigned I/O
E080-8F	SEL 8
E090-9F	SEL 9
E0A0-AF	SEL A
E0B0-BF	SEL B
E0C0-CF	SEL C
E0D0-DF	SEL D
E0E0-EF	SEL E
E0F0-FF	SEL F
E100-E2FF	DRAM
E300-E3FF	SAMBUG09 DRAM
E400-FFFF	SAMBUG09/PRAM/DRAM

Figure 1. Memory map for COB09



Dear Don,

Please find enclosed a program to calculate resistor values for National Semiconductor's three-terminal voltage regulators (like LM317). The program was taken from one of NEC's "Application Briefs" but has had some major corrections and has been adapted for Flex users.

Although many calculators could do the job, why not use the computer? It is there anyway. Since the program is written in BASIC, I am not 100% sure that it will function in all circumstances, but as far as I have tried it out, it works satisfactory.

As an avid reader of your magazine, I would like to tell you that I enjoy most of it, but... I think you should be careful not to emphasize too much on specific matters in the "Take-Away"-systems like the CoCo. We are still, I hope, a lot of users building our own systems from scratch or using boards, and as one of those, I like most of all articles describing issues of more general interest, the contribution on Structured Assembler in the May issue is one of the best and most interesting I have read. As a striking contrast is in the same issue used four full pages on an obscure cross reference listing. I think the space is too valuable for that.

Hopefully the above remarks will cause reactions from other readers. I know it is impossible to please all, but a good discussion on what should be in '68 Micro can only make our magazine even better.

Yours sincerely,

Niels Oosten

Niels Oosten
Boutykkevej 109
DK-2650 Hvidovre
Denmark.

```
1000 PRINT *
1010 PRINT *
1020 PRINT *
1030 PRINT *
1040 PRINT *
1050 PRINT *
1060 PRINT *
1070 PRINT *
1080 PRINT *
1090 PRINT *
1100 PRINT *
1110 PRINT *
1120 PRINT *
1130 PRINT *
1140 PRINT *
1150 PRINT *
1160 DIM S(2),O(23)
1170 DIM B,2
1180 PRINT "ADJRES by Niels Oosten - 03-04-05":PRINT:PRINT
1190 PRINT "This program accommodates all three terminal adjustable"
1200 PRINT "voltage regulators manufactured by National Semiconductor."
1210 PRINT:PRINT:PRINT "Enter resistor value tolerance in percent: ";
1220 INPUT 0, T:PRINT
1230 RI=237:SI=1:IF T<=2 THEN 1300
1240 RI=240:IF T<=5 THEN 1200
1250 RI=270:SI=2
1260 IF T<=10 THEN 1290
1270 T=10
1280 IF T<3 THEN 1300
1290 FOR JI=0 TO 23:READ Y:O(JI)=Y:NEIT JI
1300 PRINT "Do you wish to assign the value of RI(Y/N)":INPUT A1
1310 FOR JI=0 TO 23:READ Y:O(JI)=Y:NEIT JI:IF LEFT$(A1,1)="N" THEN 1390
1320 PRINT "The following values are available"
1330 FOR J=0 TO 17 STEP 59:PRINT O(J):IF J<10 THEN 1350
1340 PRINT
1350 NEIT J:PRINT:PRINT "Enter resistor value selected for RI: ";
1360 INPUT 0, RI:PRINT
1370 FOR J=0 TO 17 STEP 59:IF O(J)=RI THEN 1390
1380 NEIT J:GOTO 1320
1390 PRINT "Do you wish to provide a value for R2(Y/N)":INPUT A2
1400 IF LEFT$(A2,1)="N" THEN 1430
1410 PRINT "Enter value of R2: ":INPUT 0, R2:PRINT
1420 R=1:S(1)=RI:S(2)=R2:GOTO 1540
```

'68' Micro Journal

```
1430 INPUT 0, "Enter the NOMINAL output voltage: ":V:PRINT
1440 IF V<1.25 THEN 1460
1450 PRINT "Voltage must be equal to or greater than 1.25 Volts":GOTO 1430
1460 IF V<=34-T THEN 1480
1470 PRINT "Voltage must be equal to or less than 34-T Volts":GOTO 1430
1480 S(1)=RI:GOSUB 2070
1490 FOR A1=-1 TO 2:FOR B1=0 TO 23 STEP 59:V1=10*A1+O(1):IF V1>V THEN 1510
1500 S(2)=V1
1510 IF V1<=V THEN 1530
1520 N1=V1:GOTO 1540
1530 NEIT B1:NEIT A1
1540 V3=1.25*(1.25/(S(1)-S(1)+7/100))+S(2)+S(2)+7/100
1550 V1=1.25*(1.25/S(1))+S(2)
1560 V2=1.25*(1.25/(S(1)-S(1)+7/100))+S(2)-S(2)+7/100
1570 IF V3<=V THEN 1590
1580 PRINT "The value of R2 is too large....":GOTO 1430
1590 PRINT:PRINT "R1 =";S(1);", R2 =";S(2);" ("';T;"percent values)"
1600 IF V<0 THEN 1620
1610 V4=V3
1620 IF B1=0 THEN 1640
1630 N1=S(2)
1640 IF R1=0 THEN 1660
1650 PRINT "(R2 =";N1;"ohms in parallel with";R1;"ohms)"
1660 PRINT "Vnom =";V1; ", Vmin =";V2; ", Vmax =";V3
1670 IF R1=0 THEN 1830
1680 IF O=1 THEN 1860
1690 IF S(2)>V THEN 1730
1700 PRINT:PRINT "Do you want the next higher value for R2(Y/N)":INPUT A3
1710 IF LEFT$(A3,1)="N" THEN 1860
1720 S(2)=S(2):S(2)=N1:GOTO 1540
1730 PRINT:PRINT "Do you want a trim resistor(Y/N)":INPUT A4
1740 IF LEFT$(A4,1)="N" THEN 1840
1750 P=(INT(LOG(S(2)/LOG(10)))-2
1760 FOR A1=P TO 5:FOR B1=0 TO 23 STEP 59:R1=10*A1+O(1):
1770 V3=(R1+S(2))/(R1+S(2))
1780 IF V3>V THEN 1810
1790 V4=V3:R1=R1
1800 IF V4<=V THEN 1820
1810 S(2)=V4:R1=R1:GOTO 1540
1820 NEIT B1:NEIT A1
1830 IF V3>V4 THEN 1860
1840 PRINT:PRINT "+++ Your original resistor values were better!! +++"
1850 S(2)=S(2)
1860 PRINT:PRINT "Do you wish to add a switched resistor(Y/N)":INPUT A5
1870 IF LEFT$(A5,1)="N" THEN 2060
1880 PRINT "What is the voltage drop of the switch?":INPUT S2
1890 IF S2<=1 THEN 1910
1900 PRINT "Maximum drop of 1 Volt allowed":GOTO 1880
1910 PRINT "What new NOMINAL voltage is required?":INPUT V5
1920 IF V5<1.25*S2 THEN PRINT:PRINT "R3 is ZERO":GOTO 2040
1930 IF V5>=1.25*S2 AND V5<=V2 THEN 1950
1940 PRINT "Can't be done.....":GOTO 1910
1950 L1=0
1960 L1=1.25/S(1):N3=(V5-1.25*S2)/(L1-(V5-1.25)/S(2))
1970 FOR A1=-2 TO 4:FOR B1=0 TO 23 STEP 59:V4=10*A1+O(1):IF V4>=N3 THEN 1990
1980 L1=V4
1990 IF R3>=V4 THEN 2010
2000 L2=V4:GOTO 2020
2010 NEIT B1:NEIT A1
2020 R4=L1:GOSUB 2080:L3=S1:R4=L2:GOSUB 2080:L4=S1:PRINT
2030 PRINT "R3 =";L1;"for";L3;"Volts, or":PRINT "R3 =";L2;"for";L4;"Volts."
2040 PRINT:PRINT "Do you wish a different switched resistor(Y/N)":INPUT A6
2050 IF LEFT$(A6,1)<>"N" THEN 1910
2060 END
2070 Y=1:(V/1.25-1):RETURN
2080 S1=1.25*(S(2)+R4+S(1)+R4+S(1)+S(2)+S(2)+S(1)+S(2))/(S(1)+S(2)+R4+S(1))
2090 RETURN
2100 DATA 100,110,121,133,147,162,178,196,215,237,261,287
2110 DATA 316,348,383,422,464,511,562,619,681,750,825,909
2120 DATA 100,110,120,130,150,160,180,200,220,240,270,300
2130 DATA 330,360,390,430,470,510,560,620,680,750,820,910
```

Dear Mr. Williams,

Being the lazy person I am, I don't like to type anymore than I have to. Therefore entering the data on every letter and name I write you something as to avoid. Since I use TSC's editor and text processor I decided to have the system enter today's date for me. The following macro definition will print the system date info identified on a printed anything it is called with the command .BT from TSC's text processor.


```

**
*OR DT
*IF AN-1 .IF AN-1 .TL DATE January 30, 1987**
*IF AN-1 .IF AN-2 .TL DATE February 30, 1987**
*IF AN-2 .IF AN-3 .TL DATE March 30, 1987**
*IF AN-3 .IF AN-4 .TL DATE April 30, 1987**
*IF AN-4 .IF AN-5 .TL DATE May 30, 1987**
*IF AN-5 .IF AN-6 .TL DATE June 30, 1987**
*IF AN-6 .IF AN-7 .TL DATE July 30, 1987**
*IF AN-7 .IF AN-8 .TL DATE August 30, 1987**
*IF AN-8 .IF AN-9 .TL DATE September 30, 1987**
*IF AN-9 .IF AN-10 .TL DATE October 30, 1987**
*IF AN-10 .IF AN-11 .TL DATE November 30, 1987**
*IF AN-11 .IF AN-12 .TL DATE December 30, 1987**
**

```

Please feel free to publish this letter. I hope your readers will find this memo useful.

Sincerely,

Roger O'Saughnessy

Roger O'Saughnessy
2865 SW 124 Ave.
Beaverton, Oregon 97005
(503)444-9488

Ronald W. Anderson
3340 Sturbridge Ct.
Ann Arbor, MI 48105

July 10, 1983

'68' Micro Journal
P.O. Box 849
Mixon, TN 37343

Attention: Don Williams Sr.

Dear Don,

Please publish this in conjunction with my column in the earliest possible issue.

I've done OmegaSoft a great injustice. I've been away from using their compiler for a few months, and it seems that I inadvertently used my disk containing their version 1, which didn't have BYTE variables, for my tests reported in the July issue. Bob Reissler called me to tell me that my byte counts were way off for version 2. That comparison of output for "TABLE2 (Y+3) - TABLE2(Y+3) - 11" had OPS down for 62 bytes. Bob tells me that the correct number is 32, right in there with PL9 and Introl. Of course my overall byte count is wrong too. I'll run the test over again and report the accurate results with the latest version of OPS Pascal.

You have a column already in the mill that I wrote a month or two ago, regarding the BYTE price benchmark in which I reported that I had verified the BYTE reported time of 40 seconds at 1 MHz and 20 at 2 MHz for OmegaSoft Pascal. I must now report that those results too, were with the old version. Bob Reissler tells me that the current version at 2 MHz runs the benchmark in 12 seconds. That means that we have Mindrush C, Dynasoft C, Introl C, OmegaSoft Pascal, and PL9 running that benchmark in 10, 10, 11, 12, and 13 seconds respectively (and approximately).

Bob, I'm truly sorry I published incorrect information. I'll rerun the tests and report verification of your times. I intend to update the table of times for the BYTE benchmark now and then. When I do that I will publish the whole table with any corrections that become necessary.

Yours truly,

Ron Anderson

Ron Anderson

WATSON/BRADY HOME ACCOUNTING PROGRAM SYSTEM
COMPOUND ENTRY UPDATE

When published in July, 1982 by '68' Micro Journal (Volume IV, Issues VII through XI), one of the shortcomings of HAPS was its inability (at that time) to accept compound journal entries--entries where there is more than a single debit account and a single credit account.

The importance of being able to make compound journal entries becomes apparent when you need to write a single check for more than one category of expenses (such as gasoline and household expenses). Under the current version of HAPS this created a problem, with the resulting credit entries totaling the amount of any such check, but not ever showing the actual amount of any such check in one place. A similar situation occurs when a salary check, with several deductions, is received and recorded in HAPS. We are happy to report, as requested by several users of HAPS, this problem has been eliminated.

We have now been able to implement compound entries in HAPS, and the modifications ("mods") for that purpose are set forth below, indicating which of the HAPS programs should be modified and how. Since these mods are of a much newer 'vintage' than was HAPS when it was published, there can be no assurance that the mods are bug-free. Several users indicated they needed such a feature, however, so we have elected to 'go to press' immediately and not wait until the mods have been used extensively before submitting them for publication. The mods have been used on both of our home systems, however, without any problems. Please let either of us know if there are any user problems with the mods.

42

The mods keep the basic data-base structure of HAPS intact, a requirement for all of us who insist on being able to access the records of prior periods. Accordingly, the familiar 'Dr., Cr., Check #, Date, Payee, and Amount' format of the monthly and yearly transaction files remains the same. New additions you will note when using the mods are a 'dummy account' (9999) appearing opposite genuine debit and credit accounts when utilizing the compound entry mode of entering transactions. This dummy account will be ignored by the processing programs, so you won't be seeing it on your monthly general ledger printouts. It will be in the transaction data-bases, however, and its appearance indicates that the entry had more than a single debit and single credit.

The way in which the mods are structured will put all such entries together in the transaction data-bases, and they will all bear the same check/item number, which will also serve to link them together for future reference.

In order to make a compound journal entry, while in the entry program (TENTRY.BAS), it will prompt you to enter a 'C' prior to making such an entry. You can then enter up to 5 debit entries (accounts and their amounts). You can begin making offsetting credit entries by again entering a 'C'. When the total of all credit entries equals the total of all debit entries, the program will permit you to enter other needed information, such as check number, date, and payee. The program mods have been error-trapped to prevent any compound entry from being out of balance, but, should this occur, the posting program (POSTGL.BAS) will continue to discover the out-of-balance condition and abort.

As indicated in the description of HAPS (July, 1982 issue of '68' Micro Journal), it is not a substitute for a basic understanding of double-entry bookkeeping. With the mods, however, it is now a full-fledged general ledger system. We hope you enjoy using it.

Ernest Steve Watson
11701 St. Charles
Little Rock, AR 72211

F. Dale Brady
7729 Bradley
Little Rock, AR 72209

MAKE THE CHANGES INDICATED TO PROGRAMS OF THE W/8 HAP SYSTEM TO IMPLEMENT COMPOUND JOURNAL ENTRIES. ALL LINES ARE NEW AND TO BE ADDED, UNLESS INDICATED OTHERWISE.

CHANGES TO 'TENTRY.BAS'
165 DIM D(5),D(5),D(5)
166 DIM C(5),C(5),C(5)

405 PRINT"ENTER 'C' AND A CARRIAGE RETURN FOR A COMPOUND ENTRY"
407 PRINT"MAXIMUM OF FIVE DEBITS AND FIVE CREDITS"

442 IF AN\$="C" THEN 1000

1000 REM START OF COMPOUND ENTRY ROUTINES
1010 FOR I=1 TO 5:REM MAY BE INCREASED
1020 INPUT "Enter ACCOUNT to be Debited",ANS

1930 REM ADD TO FILE SIZE
1940 GET#1,RECORD#1:FIELD#1,2ASIS
1950 LSETTS=CVT\$ (TN)
1960 PUT#1,RECORD#1
1970 RETURN
1980 FOR J=1 TO X
1990 TN=TN+1:TA=TA/5:SA=TA-(TA/5)
2000 IF SA=0 THEN FIELD#1,2ASIS:LSETTS="":PUT#1,RECORD#1+1
2010 GET#1,RECORD#1+1
2020 FIELD#1,SA=50ASIS,2ASTOS,2ASTOS,2ASTOS,2ASTOS,26ASTPS,8ASTAS
2030 LSETTS=CVT\$ (999)
2040 LSETTS=CVT\$ (C(J))
2050 LSETTS=CVT\$ (CN)
2060 LSETTS=VS
2070 LSETTS=PS
2080 LSETTS=CVT\$ (C(J))
2090 PUT#1,RECORD#1+1
2100 GOSUB 1930
2110 NEXT J
2120 BOTO 170
2130 END

CHANGES IN OTHER HAP SYSTEM PROGRAMS

CHANGE TO 'EDIT.BAS'
695 IF D(=999 GOTO 710

CHANGE TO 'EDITGL.BAS'
695 IF D(=999 GOTO 710

CHANGE TO 'POSTGL.BAS'
465 IF D(J)=999 GOTO 480

475 IF C(J)=999 GOTO 490

500 (ADD 'PRINT' BEFORE "HIT ANY KEY...")

CHANGES TO 'CHECKMTR.BAS'

305 C=C+CVT\$(TC)
310 (CHANGE LINE TO READ AS FOLLOWS:
'IF D=CVT\$(TN) AND C=101 THEN 350')

1030 PRINT
1040 IF AN\$="C" GOTO 1180
1050 FN=VAL(ANS)
1060 FOR J=1 TO X
1070 IF FN=H(J) GOTO 1100
1080 NEXT J
1090 PRINTCHR\$(71);ACCT\$;P\$;"NOT FOUND (REENTER)".BOTO 1010
1100 D(=AS(J),D(=H(J)
1110 INPUT "Enter AMOUNT of Debit Entry",ANS
1120 D(=VAL(ANS)
1130 PRINT
1140 TD=TD+D(=)
1150 PRINT "Enter 'C' for Credit Entries"
1160 NEXT I
1170 PRINT
1180 PRINT"TOTAL OF DEBIT ENTRIES = \$ ";TD
1190 FOR K=1995

```

1200 INPUT "Enter ACCOUNT to be Credited",ANS
1210 P=VAL(ANS)
1220 FOR J=1 TO X
1230 IF P=H(J) GOTO 1260
1240 NEXT J
1250 PRINTCHR$(71); " CCR":P;"NOT FOUND (REENTER)";GOTO 1010
1260 C$(R)=AS(J);C$(R)=H(J)
1270 INPUT "Enter AMOUNT of Credit Entry",ANS
1280 C(R)=VAL(ANS)
1290 TC=TC+C(R)
1300 IF TC=0 GOTO 1360
1310 PRINT "TOTAL OF CREDIT ENTRIES = $ ";TC
1320 PRINT
1330 IF TC>0 THEN PRINT "ENTRY OUT OF BALANCE";TC=0;GOTO 1010
1340 NEXT R
1350 IF TC<0 THEN PRINT "ENTRY OUT OF BALANCE";TC=0;GOTO 1010
1360 INPUT "Enter check number",CNS
1370 PRINT
1380 INPUT "Enter payee/source",PS
1390 PS=LEFT$(PS,26)
1400 IF PS="" THEN 1450
1410 PRINT "DATA (RETURN IF CORRECT) = ";PS
1420 PRINT "ELSE ENTER NEW MONTH ";
1430 INPUTLINEANS
1440 IF ANS="" THEN 1500
1450 INPUT "ENTER DAY (1-31)",ADS
1460 IF VAL(ADS)<1 VAL(ADS)>31 THEN 1450
1470 IF LEN(ADS)>2 THEN ADS="0"-ADS
1480 IF PS="" THEN YS="0"-ADS;GOTO 1410
1490 YS=ANS+"-"+ADS
1500 P=INTCNS
1510 PRINTTAB(15); "ACCT. DR./CR.";
1520 P=INTCNS; "AMOUNT DR.";
1530 PRINTTAB(50); "AMOUNT CR."
1540 PRINT:PRINT
1550 FOR J=1 TO 18-1
1560 PRINT D$(J); " ";D$(J);TAB(37);
1570 PRINT D$(J)
1580 NEXT J
1590 PRINT
1600 PRINT "Total Debits = $ ";TAB(37);TC
1610 PRINT:PRINT
1620 FOR J=1 TO X
1630 PRINT C$(J); " ";C$(J);TAB(52);
1640 PRINT C$(J)
1650 NEXT J
1660 PRINT
1670 PRINT "Total Credits = $ ";TAB(52);TC
1680 PRINT
1690 TC=0;TC=0
1700 PRINT "Check/Item = ";CNS
1710 PRINT
1720 PRINT "Payee/Source = ";PS
1730 PRINT
1740 PRINT:PRINT "IS THIS CORRECT (Y/N)";
1750 PS=INCHS(0)
1760 IF PS="N" THEN 1770
1770 IF PS<>"Y" THEN 1740
1780 FOR J=1 TO 18-1
1790 T=T+1;R=R+1;S=S+1;R$(R)=R$(S)
1800 IF S=OTHER FIELD;25AS25;LSETTS="";POT=1;RECORDR=1
1810 GET=1;RECORDR=1
1820 FIELD=1;S=1;SAS15,2A TDS,2 STCS,2ASTMS,1DASTYS,26ASTPS,BASTAS
1830 LSETTS=CVT$(0)(J)
1840 LSETTS=CVT$(999)
1850 LSETTS=CVT$(CNS)
1860 LSETTS=YS
1870 LSETTS=PS
1880 LSETTS=CVT$(D$(J))
1890 PUT=1;RECORDR=1
1900 GOSUB 1930
1910 NEXT J
1920 GOTO 1980

```

Re: Home Accounting Program

Dear Don:

This letter is written to you in some haste because of an error I just found in the "Compound Entry Update" I sent you recently. Its one of those 'subtle' errors which can slip by when all conditions have not been fully tested.

In short, if you can still make corrections to the listing which I submitted, the following lines should be changed; if not, please publish this letter with my apologies to your readers.

ADDITIONAL CHANGES TO 'ENTRY.BAS'
 IN ORDER TO MAKE COMPOUND ENTRIES CORRECTLY POST

```

1040 IF ANS='C' THEN IS=IS-1;GOTO 1100

```

```

1350 FOR J=1 TO 18

```

```

1780 FOR J=1 TO 18

```

The foregoing changes will prevent the program from incorrectly posting an unbalanced compound journal entry to the transaction file. This condition, prior to its correction, would only occur when the maximum number of debit entries (five) had been entered. Any lesser number of debit entries would be posted correctly.

Again, sorry for any inconvenience which this error may have caused you or your other readers.

Sincerely,




'68 MICRO JOURNAL
 ATTN: Don Williams
 5900 Cassandra Smith
 P.O. Box 849
 HIXSON, TN
 37343 U.S.A.

Worstead Laboratories (Reg. Office)
 North Walsham, Norfolk NR28 9SA
 Tel: 0692 405189
 Telex: 97360 SHARET G

Your name

Our Ref: WND/pd 089

Date: 5/83

Dear Don:

Further to our telephone conversation yesterday,

I thought I'd direct this to your other 'hat'.

I wish to confirm our intention to increase our advertising in '68 to a full page. I also wish to take out a similar size advert in Ron Anderson's book.

Send us the forms!

I also wish to remind you and your readers what our policy in software support is:

If ever a software or hardware product bought from us does not perform to our claimed specifications we will fix the bug free of charge. We don't care when, where or how the bug is discovered, whether it was the day he got it or two years later. If this means giving the customer the latest version of the product free of charge, he will get the latest version of the product free of charge.

The point is that we have sold him something that did not meet our claims. The time he discovers it relative to when he bought it has absolutely no bearing on our moral responsibility to ensure that he gets what we claimed he would get.

Providing the customer registers his copy with us he will AUTOMATICALLY be sent notice of any bugs reported to us and how they are fixed with binary patches if possible. If it is not possible to fix the bug with a binary patch he will be asked to send his original disk to us. We will then return it to him free of charge, all he is out of pocket is a bit of his time and the postage.

If a customer has bought an early version of any of our software products and wants the ENHANCEMENTS (as opposed to fixing bugs, which are free) of the latest version we will supply an upgrade disk (and usually a new manual) for \$25.00. All we ask is that the ORIGINAL disk be returned to us. This offer is applicable to any of our products at any time. We don't take ten cents an upgrade. The \$25.00 barely covers the cost of the time it takes to make a new disk, the cost of the manual, and the cost of air mail postage to anywhere in the world.

SUPPORT WHAT WE SELL!

Let me take this one step further. We are excited that anyone can get away with selling you a piece of software with known bugs in it, and just because you don't discover the bugs until after your 90 day warranty runs out, you have to buy the latest version of the product at the current price. We don't think that policies like this make any commercial sense nor do we think that they are very moral but policies like this seem to be commonplace.

HOW DO PEOPLE GET AWAY WITH IT I'D LIKE TO KNOW!

Yours faithfully,

William E. Dickman
 DIRECTOR

Editor's Note: The above letter speaks for itself. Wouldn't it be nice if all the Standard S50 Bus vendors both software and hardware adopted the same attitude. I wonder how much larger our user base would be? I have watched Windrush and am impressed with not only their policies, but their continuing growth in hardware as well as software products. As I stated in an 'Ed Note' in a recent FLEX USER NOTES column by Ron Anderson, "I have heard nothing but good things about WINDRUSH and their products and company policies."

Bob and Tom, over at the SOUTH EAST MEDIA Division, have both told me how delighted that they are now able to offer immediate delivery of WINDRUSH software products. No waiting for overseas deliveries (and the increased postal rates), and they tell me they have the latest versions in stock, ready to go. And it is certainly nice to know that the user will be assured that he/she will not be burdened with additional expense, if any late-coming bugs are discovered. Our thanks to the fine folks at WINDRUSH for their exceptional warranty attitudes and policies.

DMW

**SUPPORT YOUR
 ADVERTISERS**

Computer Systems Consultants, Inc.
1456 Latta Lane, Congress, GA 30207

E. A. (Bud) Pass, Ph. D.

Several items in the Bit Bucket and several articles in '68' Micro and in other magazines have provided programs or patches for files, in particular, one provided an enhanced print routine. Apparently, many people are unaware of the differences among the versions of flex on the 6800. There are several primary versions and many minor ones, each of which has its own peculiarities. Although they are all descended from the General Flex sold by Technical Systems Consultants, most of the implementations have enhanced the original design to better support one company's hardware. Several representative versions are as follows:

TSC for SWTPC
TSC for Esorcisor
GIRIX
SWTPC
Data Comp Color
Frank Hogg Color
Spectral Color

The original TSC flex has the following limitations overcome in the SWTPC, GIRIX, and other implementations:

Printer driver resides in holes in flex or in user memory (SWTPC provides RM Area for printer drivers).
Disk drivers support one type of interface only (not both DMA and PIO, as provided by GIRIX and SWTPC).
TSC SWTPC disk drivers do not support double density or double track 5.25" disk drives.
Extended memory is not supported (SWTPC recognizes it).
Case switch is modifiable only by program logic.
No fast disk copy (SWTPC provides RIRORR, GIRIX provides BACKUP).
No object file modification program provided.
(SWTPC provides FIX, Data Comp provides MEMEX and DISKEX).

Printer driver differences probably cause the most confusion among the versions. The SWTPC printer drivers load into the utility space and relocate themselves into an area reserved for them by the RM command. This has the advantage of not requiring RENAME to be moved, which is disastrous to some programs, such as TSC BASIC. SWTPC flex does not use the PRINT.SYS program to set up the printer driver, as P.CMD or S.CMD performs all required set up. In order to remain compatible with such programs as TSC BASIC which explicitly load PRINT.SYS, P.CMD or S.CMD may be copied to a name as PRINT.SYS on the system drive. Then the desired driver may be loaded, on either system, under the name PRINT.SYS.

Sincerely,



E. A. (Bud) Pass

press release

Contact: Don Sinkiewicz
Technical Systems Consultants, Inc.
111 Providence Road
Chapel Hill, North Carolina 27514
(919) 493-1451

FOR IMMEDIATE RELEASE

UNIFLEX™ BASIC 68000 FOR THE UNIX™ OPERATING SYSTEMS

IS MODELED AFTER DEC'S BASIC-PLUS

Chapel Hill-- Modeled after DEC's BASIC Plus, UNIFLEX BASIC 68000 runs under the multi-user, multi-tasking UNIX™ Operating Systems and is available for OEM licensing. Technical Systems Consultants, Inc., which was established in 1976 and is the oldest company writing system software exclusively for the Motorola family of microprocessors, has designed UNIFLEX™ BASIC specifically for the microcomputer environment. The interpreter is written entirely in assembly language and has several unique features which support and take advantage of the UNIX Operating Systems.

Simple features include access to system time and date, access to the running task number, and access to the calling terminal number. Some UNIX Operating Systems provide for automatic record locking which permits two users to access the same data base without conflicts. "Shared text", a feature of some UNIX Operating Systems, allows several users to share a single copy of the BASIC interpreter, thereby saving considerable system memory. UNIFLEX™ BASIC 68000 supports both these features if they are supported within the UNIX Operating

UNIX™ is a trademark of Bell Laboratories.

UNIFLEX™ is a trademark of Technical Systems Consultants, Inc.

The floating-point math routines contained in UNIFLEX BASIC provide 16.8 digits of precision. The built-in math functions are accurate to a minimum of 13.5 digits, with most accurate to 16 digits. An "approximately equal to" operator can be used to compare floating-point values when there is a chance that round-off error has perturbed the values. Two floating-point values are considered approximately equal if their difference is small compared to their respective values. The user specifies just what small means. Integer variables, which are also supported, provide for speed in control loops and array indexing.

UNIFLEX BASIC allows file sizes up to one billion bytes and supports three types of files: sequential, record I/O, and random files accessed by virtual arrays. A sequential file is one in which the data must be read in the order that they appear in the file. The "print" command is used to write data to a sequential file just as it is used to write to a terminal. The "position" command allows a BASIC program to position a sequential file to any character within the file. The most frequent uses of this statement are (1) to rewind the file (start from the beginning again), (2) to find the end of the file so that additional data can be appended to the file, and (3) to return the current position in a file so that the user can find the position later. Record I/O files contain data that are stored on disk in records of fixed length. The length of each record may range from 1 byte to 16,383 bytes. Any record in a record I/O file may be randomly read or written on request. The data in each record are easily defined as ASCII characters, binary numeric data, or a combination of the two. Virtual arrays allow a program to store a data array in a disk file rather than in memory. Thus, the array can be much larger than available memory, and the data in the array remain on disk after the program terminates and can be used at a later time by another program. The methods of accessing the data in a virtual array and in a standard memory array are exactly the same.

BASIC contains several features which allow easy interfacing with the UNIX Operating Systems. The "exec" statement allows the BASIC programmer to call on another program of the UNIX Operating System from an executing BASIC program. When the called program is complete, its termination status is made available to the program which invoked it.

A single command to the UNIX Operating Systems can load BASIC, load a specified BASIC program, and immediately begin execution of that program. This is accomplished by specifying the file name of the program in the same command that calls BASIC. In such a case, arguments may be specified after the file name. These command-line arguments are accessible from BASIC as an array of strings.

During the execution of a program, BASIC may detect an error in the program or data. Normally, BASIC prints both the error number and the offending line number and then terminates execution of the program. In some cases, however, the user may wish to do something else when an error is detected. BASIC provides a mechanism for trapping errors known as the "on error goto" statement. Two read-only variables indicate which error was detected and the line number in which the error occurred. The user can use these variables in an error-handling routine which specifies what actions to take in response to particular errors.

A "compile" command allows BASIC to save programs on disk in a concise form whose source cannot be recovered. The resultant saved program is, in most cases, smaller than the same program saved with the "save" command, and it loads into memory faster.

The UNIFLEX BASIC line editor allows the user to modify an existing line without having to retype the entire line. If, during execution of the "load" command, BASIC detects an error in a line, it transforms that line into a "run" statement so that the user can later correct with the line editor.

UNIFLEX BASIC 68000 was originally implemented to run under UNIFLEX, an operating system modeled after the UNIX Operating Systems and specifically designed for the microcomputer environment.

Additional information about UNIFLEX™ BASIC 68000 is available from Don Sinkiewicz, Director of Marketing, Technical Systems Consultants, Inc., 111 Providence Road, Chapel Hill, NC 27514 (919) 493-1451.



MICROWARE

PRESS RELEASE

June 1, 1983
For Immediate Release
Contact: Ken Kaplan 313-278-8844

MICROWARE INTRODUCING 6809 C COMPILER

A C language compiler for the Motorola 6809 microprocessor has been introduced by Microware Systems Corporation. The compiler conforms to the full Barnighan and Ritchie C specification and produces optimised 6809 assembly language source code.

The C compiler runs under Microware's popular OS-9 real-time multitasking operating system, which is distributed under license by most of 6809-based computer manufacturers. The C compiler and its library give OS-9 users C-source code level compatibility with Unix application software. Microware also offers a cross-compiler version which runs under Unix on PDP-11 computer systems. The output of either version can be run on Microware's OS-9 operating system or used on stand-alone systems, including ROM-based systems.

A unique feature of the Microware C compiler is its real-time profiler capability. When activated, the profiler counts procedure invocations during program execution. A report printed after the program runs gives a statistical breakdown of function execution frequency. Using this information the programmer can identify which functions can most profitably be optimised.

A relocating 6809 assembler, linker, and comprehensive standard function library is included in the compiler package. The standard library includes all C standard functions plus Unix and OS-9 system calls.

The C compiler package is available now from manufacturers and distributors of OS-9 based computers or directly from Microware.

Microware Systems Corporation, 5835 Grand Avenue, Box 465, Des Moines, Iowa 50304, 515-278-8844

Iowa Mountain Software

BOX 212 HIAWATHA, IOWA 52233

June 3, 1983

Computer Publishing Center
68 Micro Journal
5900 Cassandra Smith
PO Box 849
Mixon, TN 37343

Dear Don,

In your June 83 issue one of your readers in an article compared the Motorola 6809 and the Zilog Z80 microprocessors referred to benchmark timings for IMS Pascal. The byte magazine article from which he got those figures did not mention that IMS Pascal is really Iowa Mountain Software's product. The current OS-9 Pascal is a similar product and users can expect comparable performance.

All inquiries about 6809 Pascal products should be directed to Microware Systems Corporation, 5835 Grand Avenue, Des Moines, Iowa 50312. Inquiries about Motorola 68000 and National 16032 versions of Pascal, LISP, and operating systems should be directed to Iowa Mountain Software.

Sincerely,

Michael Winble
President

Don Williams
'68' Micro Journal
P.O. Box 849
5900 Cassandra Smith
Mixon, TN 37343

Dear Don,

I am writing to express my thanks to you and the people involved in producing the '68' Micro Journal for such an informative magazine.

Here in Australia we are not lucky enough to have the wealth of information about the 68XX products and software that must be available in the US. I have had many frustrating and enjoyable hours working on my Pennywise 6809 computer. Lately, I have been dabbling in the joys of Pascal

'68' Micro Journal

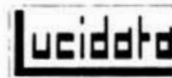
programming. I recently purchased a copy of an excellent book by Alan R. Miller called PASCAL PROGRAMS For Scientists And Engineers. It contains many interesting and thought provoking programs. The only problem I experienced was finding the time to type in all the programs. If any of your readers have purchased this book and would like a copy of the programs I would be happy to supply them on either a 5.25 or 8 inch disk for a nominal charge. The charge being to cover the cost of a diskette and postage.

Also, if any of your readers would like to communicate with a fellow 68XX devotee I would be happy to reply to their letters. Could you please include this letter in your column.

Yours Sincerely,

ALAN MILES
7 Anderson Street
Clifton Hill
Melbourne, 3068
Australia.

To: Don Williams Sr.
'68' Micro Journal
POB 849
5900 Cassandra Smith Rd.
Mixon, Tennessee 37343



Lucidata Ltd PO Box 178
Cambridge CB2 7SE England
Telephone (0223) 841808
350010

Our Ref: PAS/0583/06
Your Ref:

Date: Thursday 28th May 1983

Dear Don,

Just a quick note on the hazards of publishing other people's benchmark results. I refer to David Harper's excellent obituary column in the June issue of 'COMPUTE' - 6809 - 280, in which he perceptively noted that a figure published in Byte as representing Lucidata Pascal's performance in testing a tailward traversal algorithm, looked out of order. Although we had seen both the original reference, we had not actually checked the Cribbath values as they were predominantly non 6809 and the benchmark was even stated to be tailored to processors that couldn't do multiplication and division very well!!!

Having now performed the exercise under controlled conditions, I enclose a set of results, verification of which by independent third parties, I would welcome. You can see that there is a considerable difference so perhaps Lucidata should be accused of letting the 6809 down by not showing Byte up in the first place. We have always believed that greater credibility is given to published reports from actual users than the sometimes suspect claims of manufacturers. Hence we rarely get involved in publishing 'benchmarks' ourselves.

However, for the entertainment of your readers and possible embarrassment of some computers, I am in the process of producing a 'real' benchmark program in a form that can be clearly reproduced in these pages. The program which is very heavy on processing, involves repeated matrix inversion and evaluation of determinants. Apart from measuring speed, it clearly shows the built up of errors due to the finite precision of real arithmetic. Clearly it's no good computing fast if the results don't mean anything. Dave Gitter and I used this program as one of the tests in the evaluation of the latest machines of two well known computer companies. I think Lucidata Pascal will fare well in any comparison but more importantly I hope it will demonstrate to your readers that simplistic benchmarks usually yield simplistic comparisons.

Yours sincerely,

Dr. Nigel W. Barnes
Technical Director, Lucidata Ltd.

Checked 1/2/84 by R. G. G. (R. G. G. is a registered trademark of the company)

Interchangeable Slave program as published in Byte Jan 1983 (p.283) compiled and run with Lucidata Pascal 3.1.7/8 on a 700K G1611 with 192K byte RAM and 1500 Winchester running UNIFILE 1.07 yielded

Compile and load	11.0 secs
Execution (10 iterations)	280.0 secs

Execution under the Lucidata 3.9.7 run time system on a 1MHz 587PC with 56K bytes RAM and running FL11 9 gave

Execution (10 iterations)	502.0 secs
---------------------------	------------

The 3% difference from a factor of two on the 100K/200K systems can probably be explained by crystal tolerance, multiplier mode on the UNIFILE system and slight differences in the run time systems.

It can be concluded that as no I/O is involved in the operation of this benchmark and the program easily fits in the memory available the main difference can be attributed to processor speed alone.

1019 Weatherdon Ave.
Winnipeg, Manitoba
Canada R3M 2B5
09 Jun 83

Computer Publishing Center
'68' Micro Journal
5900 Cassandre Smith
P O Box 849
Nixon, TN 37343
U S A

Dear Don:

The following patch corrects an annoying inconsistency in the
Flex-9 TTYSET command, version 1:

ADDRESS	OLD DATA	NEW DATA
C230	8EC2A4	8EC2A5
C2A1	4F4E04	596573
C2A4	4F565604	044E6F04

Now, when you type "TTYSET", you will see "PS=Yes" or "PS=No" for
the pause switch parameter. It's always nice to have a display
of data that indicates the correct way to enter the data.

Yours Truly



J. Gary Mills

Whitethorn,
3 Leeson Road,
North Balwyn, Vic. 3106.
AUSTRALIA.

Telephone +613-857-7128 (Home)

19th June, 1983.

Mr. D. Williams Sr.,
Computer Publishing Center,
'68' Micro Journal,
5900 Cassandre Smith,
P.O. Box 849,
Nixon, TN 37343,
U.S.A.

Dear Mr. Williams,

A Floating DP Register.

I have had a problem with several programs which crashed occasionally
over the last twelve months. The cause has finally been traced to running a
program which set the direct page register in the 6809, but not restoring it
on exit to Flex. The programs which crashed used the direct page register,
but did not set it first.

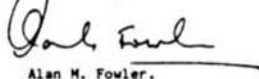
The version of Flex I am using doesn't set the direct Page register, and
leaves it up to the user's program to set it correctly. As far as I know this
applies to most versions of 6809 Flex, unless a particular company has
included an initialisation routine for the DP register when adapting Flex to
run on their system.

I have overcome the problem in two ways. Firstly, as I find a program
that uses direct page addressing, I patch it to save the DP on entry, set it
correctly, and restore it on exit. Secondly, I've put a patch into Flex to
set the DP to \$00 whenever Flex is entered by a cold or warm start. Adaptable
(General Purpose) Flex Version 3.01, and a number of other versions including
SWTP Version 2.0:3, have a call to a routine through a jump at \$DE18 which is
called for both cold and warm starts and I have modified this to include

CLRA
TFR A,DP

which sets the DP register to zero. There is probably a much better
solution, but this one at least gives me better protection against programs
that don't set the DP register correctly. As far as I know this routine is
not called when Flex is entered as a subroutine from another program, so it
offers no protection from poorly written utilities.

Yours sincerely,


Alan M. Fowler.


SMOKE SIGNAL
31336 Via Colinas
Westlake Village, California 91362

news release

SMOKE SIGNAL'S NEW VAR\68 SERIES

MICROS DESIGNED FOR SYSTEM INTEGRATORS

Westlake Village, CA. A new computer series, designed
specifically for Value Added Resellers (VARs), is being
announced by Smoke Signal as their entry into the desktop
computer market.

The new computers, called the VAR\68 series, come standard
with 128Kb of RAM, 8 serial ports, 1 parallel port and an
ergonomic video display and keyboard. The VAR\68 is now
available in configurations with: dual 750 Kb floppies;
floppy and 5Mb or 20Mb Winchester; and floppy with two 20Mb
Winchesters. Tape streamer backup is also available.
VAR\68 computers are easily upgraded. All systems can
handle 1Mb of RAM, 16 (or more) serial ports, 4 parallel
ports, special function boards--such as communications and
graphics, and hard disk storage up to 150 Mb (and higher!).
Available with the VAR\68 is OS9-Level 2, a UNIX-like multi-
user, multi-tasking operating system. Features like
password protection, record level lock-out, and dynamic



memory allocation afford the VAR maximum flexibility and
performance in multi-user applications. Resellers can
choose among a structured BASIC with Pascal type data
structures, as well as COBOL, Pascal and C compilers.
To allow a VAR to concentrate on his specialty vertical
application, Smoke Signal also offers standard accounting
software packages, word processing, the TMP family of
integrated spread sheet and data base management software as
well as a library of application software such as medical
office and CPA packages. The VAR\68 is available with a 2-4
week delivery time at end-user prices starting at \$4,325
with dual floppies, up to \$10,585 with a floppy and dual
20Mb Winchesters.

For further information, contact Don Simonsen at Smoke
Signal, 213/889-9340, or write Smoke Signal at 31336 Via
Colinas, Westlake Village, CA 91362.

Dear Don,
Having read your mag for 3 yrs now, I'd like to get in my 2 cents worth.

Instead of publishing comments or complaints about the unavailability of a piece of software—for instance a graphics editor—or even a standard screen editor—why don't you sponsor a 2 or 3 day "design seminar" for a single piece of software.

This seminar would have a prepublished agenda with discussion and the objective would be to have a design spec out line that people lived by, when the session was over.

If we could do with S.W. what other folks have done with hardware, then you'd really have something. The S-50 Bus and 6809 folk are sufficiently small in number that a good chairman/woman could pull it off.

Specs don't have to be final and chiseled in stone—only public for review and improvement, ce

Thanks for your attention

George Paquin

New Product Announcement For
COCO SLEUTH
A Program Analysis and Debugging Tool

Computer Systems Consultants, Inc.
1454 Letts Lane, Conyers, GA 30287
Telephone Number 484-483-1717/4570

Computer Systems Consultants announce the immediate availability of the COCO SLEUTH disassembler system. It is a derivative of SUPER SLEUTH, which has been available for several years for FLEX, UNIFLEX, and OS/9 operating systems. COCO SLEUTH runs under COCO DOS and processes object programs in COCO DOS format, producing COCO DOS test files.

COCO SLEUTH is a collection of three programs which enables the user to examine and/or modify binary program files on disk or in memory, on Tandy TRS-80 Color or TDP-100 or smaller computers, with at least 32K bytes of memory and at least one disk drive.

COCO SLEUTH is totally interactive, allowing the user to perform trial disassemblies multiple times, before producing the final disassembly. Programs may be disassembled from disk or memory into source code format and the source may be displayed, printed, or saved on disk for later assembly using one of several standard COCO assemblers. Address ranges may be specified as containing characters, hex constants, addresses, or instructions, in order to improve the disassembled file.

Labels produced by SLEUTH may be changed globally to labels of the user's preference. Cross reference listings of labels may be produced from any Motorola formatted assembler program, including COCO SLEUTH generated programs, to assist in debugging and modification.

Programs in ROM or on disk may be "altered" with the altered program being saved on a disk file; the resultant file could then be used to program a new ROM, etc.

COCO SLEUTH is supplied as a set of 6809 object code files for the Tandy TRS-80 Color or TDP-100 or similar computers. The processors which may be analyzed are 6800, 6801, 6802, 6803, 6805, 6806, 6809, and 6802.

COCO SLEUTH is available immediately for \$49.00.

SUPER SLEUTH for COCO FLEX is available for \$58.00 (object only) and \$69.00 (with source).

SUPER SLEUTH for OS/9 is available for \$101.00 (with source).

SUPER SLEUTH for UNIFLEX is available for \$100.00 (with source).

To obtain additional information, to request evaluation dealer copies, or to order, contact Bud Pass at the address above.



2487 WEHLE DRIVE • BUFFALO, NEW YORK 14221 • 716-631-3011

P R E S S R E L A S E

Universal Data Research, Inc. today announced Professional Video Tape production on a series of training tapes for T.S.C.'s Uniflex system and Universal Data's Database Management & Accounting Software in all popular formats.

For further information contact JoAnne Heckeen at 716-631-3911.



2487 WEHLE DRIVE • BUFFALO, NEW YORK 14221 • 716-631-3011

P R E S S R E L A S E

Universal Data Research, Inc. today announced with the overwhelming response to their Radio Shack Color Computer Software and in conjunction with the Radio Shack release of a production 64K machine all this software will be drastically reduced effective September, 1983 giving a strong business potential to the 6809 processor.

Look for our new ad in this issue!

DYNAMITE+™

"THE CODE BUSTER" disassembles any 6809 or 6800 machine code program into beautiful source

- Learn to program like the experts!
- Adapt existing programs to your needs!
- Convert your 6800 programs to 6809!
- Automatic LABEL generation.
- Allows specifying FCB's, FCC's, FDB's, etc.
- Constants Input from DISK or CONSOLE.
- Automatically uses system variable NAMES.
- Output to console, printer, or disk file.
- Available for all popular 6809 operating systems.

FLEX™ \$100 per copy; specify 5" or 8" diskette.

OS-9™ \$150 per copy; specify 5" or 8" diskette.

UNIFLEX™ \$300 per copy; 8" diskette only.

For a free sample disassembly that'll convince you DYNAMITE+ is the world's best disassembler, send us your name, address, and the name of your operating system.

Order your DYNAMITE+ today!

See your local DYNAMITE+ dealer, or order directly from CSC at the address below. We accept telephone orders from 10 am to 6 pm, Monday through Friday. Call us at 314-576-5020. Your VISA or MasterCard is welcome. Orders outside North America add \$5 per copy. Please specify diskette size for FLEX or OS-9 versions.

Foreign Dealers:

Australia & Southeast Asia: order from Paris Radio Electronics, 161 Bunnerong Road (PO Box 380) Kingsford, 2032 NSW Australia. Telephone: 02-344-9111.

United Kingdom: order from Compusense, Ltd., PO Box 169, London N13 4HT. Telephone: 01-882-0681.

Scandinavia: order from Swedish Electronics hk AB, Murargatan 23-25, Uppsala S-754 37 Sweden. Telephone: 18-25-30-00.

Computer Systems Center

13461 Olive Blvd.
Chesterfield, MO 63017
(314) 576-5020



Uniflex software prices include maintenance for the first year.

DYNAMITE+ is a trademark of Computer Systems Center.

FLEX and UNIFLEX are trademarks of TSC.
OS-9 is a trademark of Microware and Motorola.

Dealer Inquiries welcome.

All C Compilers are Not Created Equal!



**Compare Introl-C/6809
with any other C being offered
for the MC6809.**

**The differences may really
surprise you!**

Introl-C is a powerful software development tool designed for the professional. It supports the full language, is truly easy to use, and produces remarkably efficient object code for the 6809. In fact, code generated by Introl-C typically is only **half the size and twice as fast** as code produced by other C compilers on the market! As we said, all C compilers are not created equal. Introl-C/6809 delivers what others only promise.

Resident Introl-C compilers for:
OS9*, FLEX, and UniFLEX**, \$375.**

Cross compilers for:
PDP-11* hosts (Unix****), \$1500.**

Trademarks:

*Microware Systems, **Technical Systems Consultants,
Digital Equip. Corp., *Bell Labs.

INTROL
CORPORATION

647 W. Virginia St.
Milwaukee, WI 53204
(414) 276-2937



'68 Micro Journal
Attn: Don Williams
5900 Cassandra Smith
P.O. Box 889
Mission, TN 37343
U.S.A.

Windrush Laboratories (UK) Office
Rushmore Avenue, Rushmore, Reading, RG2 9BA
Tel: (0862) 408100
Telex: 97360 SWARL G

Your Ref:

Our Ref: WCD/sg 089

Date: 9/6/83

Dear Don,

As you know we have just completed a major revision to PL/9 and its associated documentation.

We are considering the following enhancements to PL/9, and would like to 'poll' your readers for their reactions to them:

- a. To take advantage of the BAT on SMT, 6812 and possibly 588 6809 processor boards to produce a working environment of 900K+ bytes under FLEX. The Editor, Compiler and Debugger can then work with absolutely massive files.
- b. Have the compiler produce object code that AUTOMATICALLY takes advantage of the BAT in the system to produce relocatable code up to 1 MB in size for a single program. The code thus produced would enable any part of the 1MB space to be accessed from any other part of the 1MB space and impose absolutely no restrictions on data and code area allocations.

We have been finding that the 64K is just not enough for some of the BIG control jobs we have been undertaking lately. Having to write the code for page swapping and/or BAT manipulations is not our idea of a good way to keep your blood pressure low either!

What if the code produced by the compiler took care of all of the memory management for you? You could then sit down and start writing a program and be completely oblivious to the BAT activity that would be required in a program in excess of 56K!

Taking this one step further why not let the corresponding Editor - Compiler - Debugger also take advantage of the BAT and let you have an absolutely monstrous source file of up to 1MB in size? Of course all of this activity would be completely invisible to FLEX.

If any of your readers has an interest in a product of this nature we would appreciate hearing from them. Whether we go ahead with this project or not and the timescale will be largely dependent on the response.

If you like the idea or have some ideas of your own let us know.

Yours faithfully,

William C. Dickinson
William C. Dickinson
DIRECTOR

**CONGRATULATIONS, ARTISAN!
CONGRATULATIONS, ELEKTRA!**



When Artisan Systems Corporation needed a system monitor for their dual-processor single-board computer, where did they turn? To Star-Kits, of course!

When AAA Chicago Computer Center needed a system monitor for their 6800 and 6809 computers, where did they turn? To Star-Kits, of course!

We would like to welcome these progressive manufacturers to the world of Star-Kits software, and thank them for adopting our HUMBUG monitor for their products. We have recently concluded several licensing agreements for HUMBUG, and for our STAR-DOS disk operating system, and there are more in the works. We're glad that other manufacturers think enough of our products to put them into *theirs*.

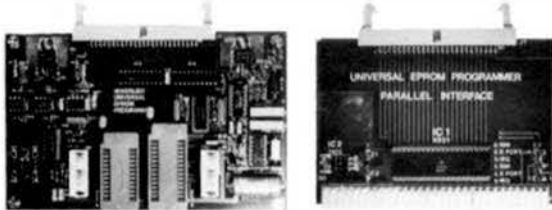
What about you? Maybe it's time for you to put our software into your hardware too. Send for our catalog.

STAR KITS

P.O. BOX 209 — J
MT. KISCO, N.Y. 10549
(914) 241-0287

WINDRUSH MICRO SYSTEMS

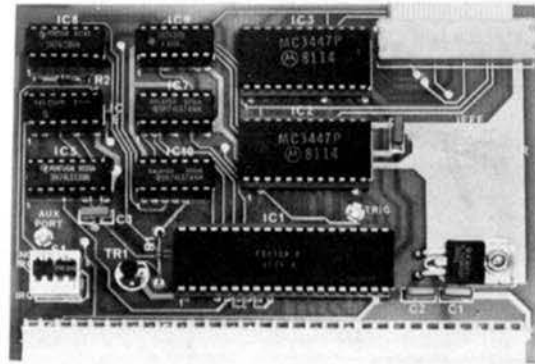
UNIVERSAL EPROM PROGRAMMER



- PROGRAMS and VERIFIES 2508, 2708, 2516, 2716, 2532, 2732A, 2564, and 2764 EPROMS. Minor hardware mods are required to program the UNIFLEX 27128.
- Tri-volt and Single Volt 2508/2708 and 2516/2716 devices are supported.
- ZIF sockets with mode selector switches eliminate 'personality modules'.
- Twin boards with five feet of twisted pair planar cable puts the programmer out on the bench where it belongs.
- SS-30 and EXORCISOR interfaces are available.
- Menu driven software provides the following facilities:
 - a. MOVE blocks of memory within the buffer.
 - b. REPS an EPROM into the buffer.
 - c. VERIFY an EPROM against the buffer.
 - d. EXAMINE and change the contents of the buffer.
 - e. PUMP the contents of the buffer in HEX and ASCII.
 - f. FILL a selected area of the buffer with a specified character.
- Software available for all versions of SSB DOS, FLEX 2, FLEX 9 and OS-9. Assembly language source files supplied on disk...enables customizing.
- Well documented users manual provides step-by-step adaptation and operating instructions.

AVAILABLE FROM GIMIX IN THE U.S.A.

IEEE-488



- SUPPORTS ALL PRINCIPAL MODES OF THE IEEE-488 (1975/8) BUS SPECIFICATION:
 - Talker
 - Listener
 - System Controller
 - Serial Poll
 - Parallel Poll
 - Group Trigger
 - Single or Dual Primary Address
 - Secondary Address
 - Talk only...Listen only
- Fully documented with a complete reprint of the XILIOBAUD article on the IEEE bus.
- Low level assembly language drivers suitable for 6800, 6801, 6802, 6803, 6808 and 6809 are supplied in the form of listings. These drivers have been extensively tested and are GUARANTEED to work!
- Single SS-30 board (4, 8, or 16 addresses per port), fully socketed, gold plated bus connectors, and IEEE interface cable assembly.

PL/9 EDITOR/COMPILER/DE-BUGGER

- Friendly inter-active environment where you have INSTANT access to the Editor, the Compiler, and the Trace-Debugger, which, amongst other things, can single step the program a SOURCE line at a time. You also have direct access to any FLEX utility and your System Monitor.
- 250 page manual is organized as a tutorial with plenty of examples.
- Fast single pass compiler produces BR of COMPACT and FAST 6809 machine code output per minute with no run-time overloads or license fees.
- Fully compatible with TSC text editor format disk files.
- Signed and unsigned BYTES and INTEGERS, 32-bit floating point REALs.
- Vectors (single dimension arrays) and Pointers are supported.
- Mathematical expressions: (*), (/), modulus (%), negation (-)
- Expression evaluators: (=), (<), (>), (>=), (<=)
- Bit operators: (&), (&=), (<<), (>>), (<<=), (>>=)
- Logical operators: (&&), (&&=), (<<=), (>>=)
- Control statements: IF..THEN..ELSE, IF..CASE1..CASE2..ELSE, BEGIN..END, WHILE.., REPEAT..UNTIL, REPEAT..FOREVER, CALL, JUMP, RETURN, BREAK, GOTO.
- Direct access to (ACCA), (ACCB), (ACCD), (CCR) and (XREG).
- FULLY supports the 6809 SWI, SWI2, SWI3, NMI, IRQ, and RESET vectors. Writing a self-starting (from power-up) program that uses ANY, or ALL, of the 6809 interrupts is an absolute snap!
- Procedures may be passed and may return variables. This makes those functions which behave as though they were an integral part of PL/9.
- Several fully documented library function modules are supplied: ISUBS, BITIO, HARDIO, HEXIO, FLEKIO, SCIPACK, STRSUBS, and REALCOM.

'... THIS IS THE MOST EFFICIENT COMPILER I HAVE FOUND TO DATE.'

Quoted from Ron Anderson's FLEX User Notes column. Need we say more?

MACE/XMACE

A co-resident EDITOR/ASSEMBLER for the 6809 written by Graham Trott which takes most of the pain out of assembly language program development:

- Friendly inter-active environment where you have INSTANT access to the Editor, the Assembler, PLIX and your System Monitor.
- MACE can also produce ASMPROC's for PL/9 with the assembly language source passed to the output file as comments.
- Includes XMACE a co-resident 6800/12/3/8 EDITOR/CROSS ASSEMBLER.

C

This is the FLEX version of the James McLoach 'C' compiler that is also available on UNIFLEX from SWTP and OS-9 from Microware:

- The FLEX implementation supports the full Kernighan and Ritchie 'C' specification except 'floats', 'doubles', and 'bit-fields'.
- Produces very efficient assembly language output with the 'C' source optionally interleaved as comments.
- Built-in optimizer will shorten object code by about 11%
- Supports interleaved assembly language programs.
- The TSC relocating assembler/linking loader (SP09-17) is REQUIRED.

MACE	(6809 FLEX only).....	\$ 98.00
PL/9	(6809 FLEX only)...(a steal at this price).....	\$198.00
'C'	Se 56K system and the TSC SP09-17 package is req'd)....	\$295.00
IEEE-488	with IEEE-488 cable assembly.....	\$298.00
SS-30	Universal EPROM programmer w/one version of software....	\$575.00
EXORCISOR	Universal EPROM programmer w/one version of software....	\$595.00
SOFTWARE	Drivers for a 2nd, 3rd, or 4th operating system.....	\$ 25.00

ALL PRICES INCLUDE AIR MAIL POSTAGE

WE STOCK THE FOLLOWING COMPANIES PRODUCTS:
GIMIX, SSB, FHL, MICROWARE, TSC, LUCIDATA, AND ALFORD & ASSOCIATES.

FLEX (tm) is a trademark of Technical Systems Consultants, OS-9 (tm) is a trademark of Microware Systems Corporation, MDOS (tm) and EXORCISOR (tm) are trademarks of Motorola Incorporated.

An SS-30C all CMOS 256K STATIC RAM board will be available SOON!

Write for details & pricing.

WORSTEAD LABORATORIES,
NORTH WALSHAM, NORFOLK,
ENGLAND. NR28 9SA.
TEL: (0692) 405189
TLX: 97360 SHARET G

GRANITE COMPUTER SYSTEMS

Route 2 • Box 445 • Hillsboro, NH 03244 • 603-464-3850

NEW PRODUCT ANNOUNCEMENT

• • TEXTWRITER II • •

TEXTWRITER used with TSC TEXT EDITOR creates a powerful word processing system. This combination enables the user to produce letters, texts, ads, labels - just about any kind of text.

TEXTWRITER II + EDITOR appear to be and function as one program. Use all of the power of EDITOR to build or modify a text file. Then use TEXTWRITER II to print it out.

A multitude of MENU driven options are provided.

Embedded control codes for full dynamic access to printer features - underlining, form feed, character size, lines per inch, line length, clear buffer, dot graphics, etc.

Left and right margins, lines per page, numbering, paging, justification, headers, footers, centering, double spacing, output entire file, segments, multiple copies, etc.

All the features of EDITOR: insert, move, copy, append, change, tabbing, find (occurrences), change, conditional change, delete, zone operations, expand, overlay, etc.

Simple to use - default parameters chosen for general use - change by menus for special needs.

Single letters execute commands or display secondary menus.

Readily move between editing and writing functions.

TSC EDITOR is normally loaded into low memory starting at \$0000. The test buffer occupies the next higher block of memory.

TEXTWRITER II is loaded above the buffer. The size of the test buffer at any time depends upon the available memory.

Matrix printers such as Epson and Okidata use control characters embedded in the text to control various options. Use TEXTWRITER II modifies EDITOR automatically. Simple patches for tape version are supplied.

Fast Efficient Position Independent Code

For S550 6809 and COCO Systems - Comprehensive Manual

FLEX (5 or 8") disc, COCO or KC Cassette \$75.00

6800-6809-68000

Programming

Turnkey Systems

CLASSIFIED ADVERTISING

Comet 80 Dot Matrix Printer \$240; ProWriter 80 column Dot Matrix Printer \$440, Both new-still in original boxes. Two CAL-COMP 8" Disk Drives \$390.00
Call Tom (615)842-4601.

For Sale: SWTPC 6809 System, Uniflex OS. Includes: 128K Ram; 20 Megabyte Hard Disk; two 8212 terminals, Dual 8" DSDD drives; Miscellaneous Software, Furniture (Desk, Stand). Less than 1000 hours on time. 1 1/2 years old. Excellent Condition. Make me an offer.
Call Marilyn (512)250-9844, Business Data Systems, 8300 Pilgrims Pl., Austin, TX 78759.

Wanted: bare SWTPC MP-A2 and MP-N boards.
Bill Marel 303-597-4575.

For Sale: SWTPC/6809 w 112K dynamic ram, Dual Qume DT-8 floppies, CDS-1 Century Data 20 Mbyte hard disk, Watanabe Intelligent plotter, Model WX4671; Software: Flex 9, Uniflex, Dynacalc, Stylograph Sp. Chkr., Basic, Pascal C and much more.
Write or call Carl Polomski, 30645 Indigo, Roseville, MI 48066, (313) 876-2010.

For Sale: SWTPC 68K 1 Meg. 6809 with 56K, Serial Ports, 2 Parallel ports, Interrupt Timer, DC-2 floppy disk controller, MF-68 with 2 5.25" SS/SD drives, SWTPC FLEX-\$810
SWTPC S/00 (S/09 minus CPU, Disk Controller and Memory) includes MP/ID and MPL2. Supports 2 megahertz clock and extended addressing-\$360.
QMMI with 256K. \$550.
TSC Software: Basic (not extended), editor, assembler, debugger, and Pascal; plus Hogg Toolkit-2, and adventure-\$350.
Peter Dibble 716-275-7507, 517 Goler House, Rochester, NY 14620

** QUALITY SOFTWARE NEEDED **

Standard S50 Bus and Color Computer

For the past few months we at the South East Media Division of Computer Publishing, Inc. (CPI), the parent company of 68 MICRO JOURNAL, have debated expanding into the software distribution business. Many other magazines have been doing so for years. Presently there are many fine examples of software that has been developed by YOU our readers, that will never see the 'light of day' unless someone, with enough exposure and willingness to continually advertise, runs with the ball.

Software is the 'backbone' for the real utilization of any computer, ours are no exceptions! Realizing that there will be some conflicts, with other advertisers, this has been no simple decision. However, since day one the foremost concern of 68 MICRO JOURNAL has been it's readers! Therefore, South East Media Division will accept, for appraisal, software that runs on 6809 systems, games, utility or applications programs.

In the past there has been too much software offered that was not quite ready, nearly, but not quite. We will strive to eliminate that element. But right up front we tell you only that we will do our very best, nothing more. Also we will strive to keep cost to a bare minimum, while securing for the author a fair return, in royalty payments, promptly paid.

Of course we will expect, no - demand, that the author keep the product free of errors (bugs), and maintain it on a prompt and business like basis. Also we shall require that authors be willing to furnish 'source' for those programs that justify, by price and utility, inclusion of same. The lack of source code, properly commented, is a continual complaint we hear. Not all programs will be sold with source, but where necessary, we will insist that it be included.

In some instances the program may be small or short and not justify itself as a 'single' sale product. In this event it will be combined with other like programs, and offered as a package. In that event the royalties will be split between the various authors.

If you have software that you feel will qualify under this program please contact the proper person as shown below.

Color Computer
Tom Williams
Bob Nay

Standard S50 Bus
Don Williams
Bob Nay

Remember, if your software has any problems or 'funnies' - GET IT STRAIGHT BEFORE YOU CONTACT US!!!!
Also get your source code in proper shape and well commented. There is too much 99% code already drifting around.

South East Media, POB 794, Hixson, TN 37343 - (615) 842-4601
A Division of CPI



Software



For Ordering Call TOLL FREE 1-800-338-6800
FLEX™ OS-9™ Color Computer

Computer Systems Consultants:

AVAILABLE NOW!!!

The Powerful Super Sleuth Disassembler for the Data-Comp, Hogg, and Spectral Color Comp, FLEX Systems;

Color FLEX Object Code only \$50.00
 Color FLEX w/ Source Code \$99.00

Normal FLEX w/ Source Code \$99.00
 UNIFLEX w/ Source Code \$100.00
 OS-9 w/ Source Code \$101.00

ALL Comp Sys Cons Programs run on the Color FLEX Systems

Great Plains Computer Co.

AVAILABLE NOW!!!

STYLOGRAPH 2.0 for the Data-Comp and Hogg Color FLEX Systems. Full screen display and editing (i.e., what you see is what you get); supports proportional printers. Operates with the 51 x 24 screens.

SPECIAL COLOR FLEX STYLO \$195.00

FLEX STYLO \$295.00
 UNIFLEX STYLO \$395.00

Fast SPELLING CHECKER; allows directly changing the Text File, adding words to the dictionary, etc. 75,000 words in less than 400 sectors (easily fits 5¼" disks with other normal commands on it also).

FLEX, Color FLEX, OS-9 \$125.00
 UNIFLEX \$175.00

MAIL MERGE — greatly extends the power and flexibility of STYLOGRAPH. Allows Multiple Text files to be printed out as one large document. Provides for merging information into the Text File during printing (such as different names and addresses), etc.

FLEX, Color FLEX, OS-9 \$145.00
 UNIFLEX \$195.00

INFOMAG Data Base Management System — An XBASIC-Based, Menu Driven, DBMS with "Built-in" Audit Tracking, Extremely Powerful Report & Format Capabilities, etc. This "Time Proven" DBMS will become the "Work Horse" of your Software "Stable."

FLEX and Color FLEX \$295.00
 UNIFLEX \$395.00

Also in Stock: Accts Rec., Accts Payable & Gen Ledger — A full blown Accounting Package that can be used together, or as separate packages; provides the IRS required Audit Tracking. (XBASIC, based on the "Osborne Business Programs.")

FLEX and Color FLEX \$295.00/PROG.
 UNIFLEX \$395.00/PROG.

Also in stock; PRIMAC; a Relocatable, Recursive Macro-Assembler and Linking Loader for the 6809. An enhancement of Ed Smith Software Works original system. Use either Motorola Format or Special Ed Smith Format. Supports Recursive Macros, Conditional Assembly, etc. All I/O thru normal FLEX System Calls. Can provide a full Cross Reference Listing. Includes a Small Line-oriented Editor as part of the Assembler. Greatly improved operating manual.

FLEX & Color FLEX 6809 PRIMAC w/Link & Editor \$150.00
 SOURCE \$ 50.00

Call for information on any requirement—

We have over 200 FLEX, Color FLEX, UNIFLEX, OS-9 and Color Computer Programs available; Games, Programming Languages, Business Software, etc., etc., etc. ??? More Info; just watch these ads!!!, write or call South East Media.

Depend On South East Media

Make Your 'BACK-UP' On South East Media

FREE DISKETTE With Each \$50 Purchase

For Ordering Call TOLL FREE 1-800-338-6800

SOUTH EAST MEDIA

5900 Cassandra Smith Rd., Hixson, TN 37343
 (615) 842-4601

*FLEX is a trademark of Technical Systems Consultants

**OS9 is a trademark of Microware

O-F

Finally the barrier has been removed from OS9™ to FLEX™ formatted disk!! Now you can READ and WRITE to a FLEX™ diskette, 5 or 8 inch, with O-F.

O-F is a new and unique program, written in BASIC09™ that performs the following functions, and comes complete with source.

1. **REFORMAT:** This module formats a disk that can be read by both OS9™ and FLEX™. Eight or five inch selectable.
 2. **FLEX.BAS:** This program does the actual read or write function to the special O-F disk. Also it has the disk format and DIR (OS9) commands. All selectable from a user-friendly menu. All selections are interactive and complete including all necessary prompts to the operator.
 3. **DIR:** This module (menu selected) allows the disk directory to be printed to the screen, while in BASIC09.
- FLEX users can read, write and use the special disk as any other FLEX disk, provided the FLEX directory is not allowed to continue beyond track zero (too many files).

\$79.95

WINCHESTER BACKUP UTILITIES

The following utilities allow the backup of any size disk system to any size diskette.

By simply inserting diskettes as requested by COPY, MULT, a large disk system (Winchester, etc) may be downloaded to your present floppy disk system, any size. No need to fiddle with directory deletions or any of the other tedious operations that must be done using a nonnal copy routine.

COPYMULT-CMD understands normal "copy" syntax and always keeps up with files already copied by maintaining directories for both host and receiving disk system, thus eliminating hours of tedious keyboard entries and other time consuming cleanup chores.

BACKUP-CMD is a special program that downloads "random" type files, any size.

RESTORE-CMD a special program to restructure copied "random" files for copying, or recopying back to the host system.

FREELINK-CMD a "bonus" utility that "relinks" the tree chain of a floppy or hard disk thereby eliminating fragmentation.

**Completely documented source files included.

**ALL 4 Programs

99.50 on 8" diskette

CHESS 6809

Requires FLEX™ and Now Runs On Any Type Terminal

Features:

- Two display boards.
- Change skill level.
- Swap sides.
- Point scoring system.
- Four levels of play.
- Stop 'Mate in 1-2-3-4' moves.
- Make move and swap sides.
- Play white or black.

\$79.95 Specify 5" or 8" disk

This is one of the strongest CHESS programs running on any microcomputer, estimated USCF Rating 1600 +.

DIET-TRAC Forecaster

DIET-TRAC Forecaster is a program that plans a diet in terms of either calories and percentage of carbohydrates, proteins and fats (C P G %) or grams of Carbohydrate. Protein and Fat food exchanges of each of the six basic food groups (vegetable, bread, meat, skim milk, fruit and fat) for a specific individual.

Sex, Age, Height, Present Weight, Frame Size, Activity Level and Basal Metabolic Rate for normal individuals are taken into account. Ideal weight and sustaining calories for any weight of the above individual are calculated. When a weight goal is given (either gain or loss), and a calorie plan is agreed upon between the computer and the individual, the number of days to reach the weight goal is projected. The starting and ending rate of weight loss is calculated, and a daily calendar with each day's weight for a 30-day period is printed.

FLEX VERSION — \$59.95

UniFLEX VERSION — \$89.95

A COLOR COMPUTER TERMINAL DRIVER

TERM is a new and long needed terminal driver for those color computer users who have developed stinging, red and watering eyeballs! TERM allows you to switch from the CoCo keyboard and TV monitor to a real CRT video terminal. Think what that means! No more 32 or even 51 characters per line, that are so blurred that you must guess as to what some characters actually are. No more squinting or cussing. Sounds great? Well TERM gets you away from all that. TERM is called from the CoCo as any other program. You then type the 'RETURN' key on the CRT video terminal keyboard and TERM configures the CRT video terminal to the proper baud rate and you are free of squinting and guessing! To return to the CoCo keyboard (God forbid), you simply type in the command EXTERM from the CRT video terminal keyboard, and you are back where you started from, squints and all.

TERM functions from Data-Comp FLEX directly and requires only a standard CRT video terminal (any ol' cheapo will do) or a deluxe terminal, either works just fine. The terminal is connected to the serial port of the CoCo by a standard cable and connector. TERM does NOT function in Radio Shack mode (must be FLEX).

If you want the luxury of 80 characters by 24 lines, or more, depending on the CRT video terminal used, then TERM is a must!

ONLY \$19.95

FREE with purchase of F-Mate

SOUTH EAST MEDIA

5900 Cassandra Smith Rd., Hixson, TN 37343

(615) 842-4601

'68' Micro Journal

SPELLB "Computer Dictionary"

No more "Let your fingers do the walking through the Dictionary" while you are inputting Text with your favorite Editor or Word Processor. **SPELLB**, written by Dan Farnsworth of PALM BEACH SOFTWARE, is more than "another Spelling Checker"; it allows you to "look up a word" from within your Editor or Word Processor so that you KNOW it is right WHEN YOU TYPE IT IN (If your Editor supports a method of passing a Command to FLEX without exiting the Program) with the **SPH.CMD** Utility (which operates in the FLEX Utility Space). Yes, it ALSO allows you to check and update the Text after you are finished; along with allowing you to ADD WORDS to the Dictionary, "Flag" questionable words in the Text for evaluation later, "View" a word in context before changing or ignoring, etc. **SPELLB** first checks a "Common Word Dictionary", then the normal Dictionary, then "Personal Word List", and finally, any "Special Word List" you may have specified. **SPELLB** also allows the use of Small Disk Storage systems.

FLEX and Color FLEX \$129.95

"JUST" Text Formatter

JUST, a "TextFormatter" developed by Ron Anderson, provides numerous features which make it a valuable addition to any FLEX Users Software Library. **JUST** is designed for formatting Text Output for Dot Matrix Printers and provides many unique features:

Output the "Formatted" Text to the Display for format analysis and change.

Output the "Formatted" Text to a Text File for use with the supplied **FPRINT.CMD** for producing multiple copies of the Text on the Printer INCLUDING IMBEDDED PRINTER COMMANDS (this Utility useful at other times also, and worth the price of the program by itself).

"User Configurable" for adapting to other Printers (comes set up for Epson MX-80 with Grafrax); provides for up to ten (10) imbedded "Printer Control Commands", such as Italics on and off, Boldface on and off, etc.

Automatic compensation for a "Double Width" printed line.

Includes the normal line width, margin, indent, paragraph, space, vertical skip lines, page length, page numbering, centering, fill, justification, etc.

Use with ANY Editor.

Supplied with "Structured Source" (Windrush PL/9); easy to see the flow of the program.

FLEX and Color FLEX \$49.95

PROGRAMMERS (both NEW and OLD-TIMERS) NOTE —

The Power and Versatility of the 6809 CPU has stretched the capabilities of the "Old Standard" Programming Practices and Procedures, and opened the door for the "Innovators" of the Industry to develop NEW and BETTER Software Development Tools. **WINDRUSH MICRO SYSTEMS** has stepped forward with three (3) new Programs that will find a home on EVERY Programmers work bench.

MACE — XMACE By Graham Trott

MACE is a combined Editor/Assembler designed to allow the Programmer to Enter, Edit, and Assemble Programs with a minimum of effort. **MACE** is designed primarily for the EASY development of small to medium sized Assembly Language programs, but larger programs can be developed using the "Spool" capabilities. The Editor (a simplified Line Editor streamlined for this package) "codes" each Op-Code, providing minimal memory requirements. **MACE** is very "forgiving", which, when combined with the "interactive" operation, makes this an EXCELLENT package for the Beginning Programmer!

FLEX and Color FLEX — \$98.00

PL/9 — By Graham Trott

PL/9 is an Editor/Compiler/Debugger all combined into ONE PACKAGE, which was devised specifically to allow the Assembly Language Programmer the "Best of All Worlds". It allows the Programmer to use "Structured Programming Techniques" while working at the Assembly Language level in a totally INTERACTIVE Program Development Cycle (just like working with BASIC: enter some code, try it, edit the code, try it again, etc.). The Single Pass Compiler supports up to 127 Character Symbols; Variable Types; Pointers; Control structures built around the "Procedure" System, IF, THEN, ELSE, BEGIN, END, WHILE, REPEAT, UNTIL structures, etc., along with Stack, A-, B-, and D-Register manipulation etc. The Editor/Assembler are similar to the **MACE** Program. The Trace/Debugger is oriented towards the **PL/9** Source Program and provides Single Stepping, Breakpointing, running a specified Line Number Range, etc. All in all, this provides an excellent Software Development Tool for utilizing the power of the 6809.

FLEX and Color FLEX — \$198.00

C — C-By James McCosh

Finally, for the "Big Time Operator", or for the beginner who can "see the future" and wants to become a part of it, here is the EXCELLENT **WINDRUSH MICRO SYSTEMS** "C Compiler". This is one of THE C Compilers for the FLEX Operating System. It can be used with normal Assemblers for most Programming, or with the TSC Relocating Assembler/Linking Loader for those "full blown" System Packages.

FLEX and Color FLEX — \$295.00

SOUTH EAST MEDIA

5900 Cassandra Smith Rd., Hixson, TN 37343

(615) 842-4601

CoCo SLEUTH(tm)
AT LAST!! A FULL-BLOWN DISASSEMBLER FOR THE COLOR COMPUTER

Computer Systems Consultants SUPER SLEUTH is a "Time Tested", reliable, PROVEN Disassembler that has gained acceptance through out the FLEX Community as an extremely POWERFUL, INTERACTIVE, Software Tool. NOW, this powerful Disassembler has been converted to run on a Standard 32K Color Computer or TDP-100 System and a Disk System. The CoCo SLEUTH(tm) Software Package consists of 3 Programs; SLEUTH (the Disassembler), CRQAM (used to globally Change Labels to a meaningful Name), and XREF (a Cross Reference Generator for Source Code Files). CoCo SLEUTH will Disassemble Disk Files of 6800, 6801, 6802, 6803 (the "Baby CoCo"), 6805, 6808, 6809, and 6502 (Apple, Atari, Commodore, etc.) Object Code if you can get it on a Color Computer Disk. (See Aug. '83 '68' Micro Journal "Color Users Notes" Column for a full Review.)

Color Computer Disk - Object Code Only \$49.00

Computer Systems Center

AVAILABLE NOW!!

DYNACALC(tm) — THE Electronic Spread Sheet for the 6809 Computer Systems. An extremely POWERFUL Business Tool, this Program will find in an unlimited number of "non-business" applications, also (for example, I have just finished setting up a Full Junior College Electronics Curriculum using DYNACALC). Advanced features like "Table Lookup" make Income Tax work easy; Column or Row Sorting for numerous applications; etc. Completely "Memory Resident", Machine Language, this Program is FAST. Utilizes STANDARD FLEX Text Files for Data, allowing the use of these Files with BASIC, Word Processors, Pascal, "C", etc.

FLEX and Color FLEX (Both FHL and Data-Comp) \$200.00
UniFLEX \$395.00

AVAILABLE NOW!!

DYNAMITE +(tm) — A "easy to use" FLEX Based 6809 Disassembler which will also disassemble 6800 Binary Files. Allows the development of a "Control File" of various Program "Boundaries" during successive disassemblies; produces Source Code that will assemble back to the original Binary Code; provides for "Label Files" which automatically replaces a Hex location with a Label Name; etc. Label Files provided for Mini-FLEX, FLEX2, FLEX9, Color Computer (for use with Color FLEX Systems), etc.

FLEX and Color FLEX \$100.00
UniFLEX \$300.00

SPECIAL!! SPECIAL!! SPECIAL!!! Only TWO (2) Left at THIS PRICE!

Star-Kits excellent SPELL 'N FIX Dictionary and WRITE 'N SPELL Word "Looker Upper" (these words are NOT in the SPELL 'N FIX Dictionary) IN ONE PACKAGE;

BOTH for ONLY \$150.00

(for FLEX and Color FLEX Systems)

When these are gone, the price goes UP!! WAY UP!! ORDER NOW!!

Also, call for "More Info" on both the FLEX Based and Color Computer Based Star-Kits Products; including the BUNBUG Monitor, Check 'N Tax Program, REMOTERM Color Computer External Terminal Program, etc.

SOUTHEAST MEDIA also carries a COMPLETE INVENTORY of both
TSC FLEX Based Software and COMPUTER SYSTEMS CONSULTANTS, INC. Software Products.

SOUTH EAST MEDIA

5900 Cassandra Smith Rd., Hixson, TN 37343

(615) 842-4601

FOR ORDERING CALL TOLL FREE 1-800-338-6800

SOUTH EAST MEDIA
5900 Cassandra Smith Rd., Hixson, TN 37343
(615) 842-4601

Software

FLEX™

OS-9™

Color Computer

Lucidata

AVAILABLE NOW!!!

PASCAL Language and P-Code Compiler, Version 3. Designed especially for Micro Computer Systems; Run-time System checks available resources for each task, allowing operation on even minimal computer systems. Based on the proposed ISO Standard. Allows linkage to Assembler Code for maximum flexibility.

FLEX and Color FLEX 5" \$190.00
FLEX 8" \$205.00

PASCAL UTILITIES — Require LUCIDATA Pascal Ver. 3. **XREF** — will produce a Cross Reference Listing of any text; aimed specifically towards Pascal Source. **INCLUDE** — allows the inclusion of other Files in a Source Text; has unlimited nesting capabilities. Also allows Binary File Inclusions. **PROFILER** — produces an Indented, Numbered, "Struclogram" of a Pascal Source Text File. Allows viewing the overall structure of large programs, and provides clues as to the integrity of the program. Supplied as Source Code; requires compilation.

FLEX and Color FLEX — Each program \$25.00

COPYCAT — Allows reading TSC Mini-FLEX, SSB DOS68, and Digital Research CP/M Disks while operating under FLEX 1.0, FLEX 2.0, or FLEX 9.0 with 6800 or 6809 Systems. COPYCAT will not perform Miracles, but, between the program and the manual, you stand a good chance of accomplishing a transfer. Includes Utilities to List Directories, Copy Files, and convert Text Files when required. Also includes a Utility for investigating Physical Compatibility problems. Programs supplied in Modular Source Code to make it easier to solve unusual problems.

FLEX and Color FLEX 5" \$50.00
FLEX 8" \$65.00

Westchester Applied Business Systems

AVAILABLE NOW!!!

XDMS Data Management System. Possibly one of the most powerful DMS's available. This machine language program is small enough to operate on a single sided 5" disk, yet provides the speed of M.L. and power limited only by the user's imagination. Supports Sequential, Hierarchical, and Random Access File Structures, and has Virtual Memory capabilities for those Giant Data Bases. Easy-to-use English Language Command Structure.

FLEX and Color FLEX \$179.95

For the "Strictly Color Computer" Users

AVAILABLE NOW!!!

Hoyt Sterns Electronics—

Intrigued by Forth???? Here is a Forth package tailored to the color computer! This package is supplied on Tape, with instructions for transferring it to disk if you wish. Written primarily in machine language, it's speed is unparalleled. A full Semigraphic-8 Editor is provided, along with "goodies" like Graphics and Sound Commands. Printer Commands, Auto-Repeat and Control Keys, etc. If you are interested in Learning Forth, a Trace Feature is provided which is invaluable. If you are a FORTH Pro, this package provides CPU carry Flag accessibility, Fast Task Multiplexing; Clean Interrupt Handling, etc. (Or; you won't "out grow" the Basic capabilities of the Implementation). Combine this package with Leo Brodie's EXCELLENT Book "Starting FORTH", and you will be a FORTH Expert before you know it (and have a lot of fun doing it!).

\$58.95

Introl

AVAILABLE NOW!!!

6809 "C" Compiler; generates extremely efficient object code. Output "benchmarks" close to 10MHz 68000 in 8 Bit Operations; 1.5 times faster than a 4 MHz Z80 when using a 2MHz 6809 System (Re. p 43, '68' Micro Journal, May '83).

FLEX, Color FLEX, OS-9 \$375.00

UNIFLEX \$425.00

For Ordering Call TOLL FREE 1-800-338-6800

One Year Maintenance. \$100.00



SOUTH EAST MEDIA

5900 Cassandra Smith Rd., Hixson, TN 37343
(615) 842-4601

™FLEX is a trademark of Technical Systems Consultants

™OS9 is a trademark of Microware

'68' Micro Journal

OS/9™, FLEX™, COLOR FLEX™, COCO™ DOS, UNIFLEX™ Software

SUPER SLEUTH DISASSEMBLER \$99-FLEX \$100-UNIFLEX \$101-OS/9

This program processes 68001, 23569, 6502 programs, enabling the user to analyze, modify, and disassemble (with updates) object code, with output to terminal, printer, and disk, and cross-reference and label-definition capabilities. Object-Only Version for Color FLEX \$50, COCO DOS \$49.

Z-80/8080/5 SUPER SLEUTH DISASSEMBLER \$99-FLEX \$100-UNIFLEX \$101-OS/9

This version of SUPER SLEUTH processes Z-80/8080/5 object code on the 68001/9.

CROSS-ASSEMBLERS each \$50.3-\$100-FLEX each \$60.5/\$120-UNIFLEX each \$55.3/\$110-OS/9

These programs and macros enable the user to process 68001, 6805, 6502, Z-80, 8080/5 programs in original format. The TSC macro assembler is required for FLEX/UNIFLEX and the OSM assembler is required for OS/9.

6805 and 6502 DEBUGGING SIMULATORS each \$75-FLEX \$80-UNIFLEX \$100-OS/9

These programs enable the user to interactively analyze, modify, and debug (i.e.) 6805 and 6502 object code.

6502-TO-6809 XLATOR SYSTEM \$75-FLEX \$80-UNIFLEX \$85-OS/9

This program enables the user to translate 6502 assembler code into 6809 assembler code, noting inexact conversions.

6800-6809 & 6809 PIC XLATORS both \$50-FLEX \$60-UNIFLEX \$75-OS/9

These programs enable the user to translate 68001 assembler programs to 6809 mnemonics and to convert 6809 programs to position-independent code and data, using PC, S, U, X, and Y as base registers.

UNIFLEX SIMULATOR FOR FLEX \$100-FLEX \$110-UNIFLEX

This program enables the user to debug UNIFLEX assembler programs using the TSC DEBUG and other facilities of FLEX.

OS/9 SIMULATOR FOR FLEX \$101-FLEX

This program enables the user to debug OS/9 assembler programs using the TSC DEBUG and other facilities of FLEX.

FULL SCREEN FORMS DISPLAY (6809 X-BASIC) \$50-FLEX \$75-UNIFLEX

These programs enable the user to define and generate table-driven full-screen display and data entry programs.

FULL SCREEN MAILING LIST (6809 X-BASIC) \$100-FLEX \$110-UNIFLEX

These programs enable the user to define and maintain mailing list-oriented data bases.

FULL SCREEN INVENTORY MRP (6809 X-BASIC) \$100-FLEX \$150 UNIFLEX

These programs enable the user to define and maintain inventories, and include hierarchical materials requirement planning.

TABULA RASA SPREADSHEET (6809 X-BASIC) \$100-FLEX \$200-UNIFLEX

These programs enable the user to generate and maintain tabular computation schemas, providing a simple user interface and sophisticated report-generation, similar to DESKTOP PLAN™ VISACORP.

TSC BASIC/XPC UTILITY PROGRAMS all \$25-FLEX \$50-UNIFLEX

These programs enable the user to resequence or cross-reference any Basic program and generate XPC Basic sort programs.

Programs in source on disk — specify size, sides, density, type, computer, OS.

Detailed printed manuals provided with all products.

For VISA and MASTER CARD give account, exp date, phone, US funds only — add 4% (10% foreign) for shipping.

Open Purchase Orders for D and B rated clients only. Call or write for catalog and dealer information.

FLEX™ Technical Systems Consultants OS/9™ Microware COCO™ Tandy

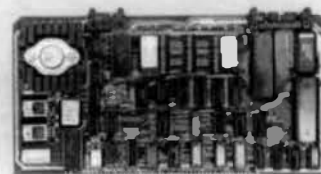
Computer Systems Consultants, Inc.
1454 Latta Lane, Conyers, GA 30207
Telephone Number 404-483-1717-4570

6809 + OS-9* + S-100

WHAT A COMBINATION!

ADS is pleased to announce support for Microware's OS-9 on the ADS 6809 S.B.C.

Used in conjunction with our Octafloppy™ disk controller card, you now have complete 6809/OS-9 capabilities for the S-100 bus. Further, the ADS Octafloppy™ gives you up to eight drives, 5¼" or 8",



single or double sided, single or double density. Expand to the power of the S-100 bus. (The bus that handles 16 bit CPU's too!) Call or write today for more information.

ads

Ackerman Digital Systems, Inc.

110 N. York, Suite 208 • Elmhurst, IL 60126
(312) 530-8992

See us at
Microware OS9 Seminar!

OS-9* Trademark of Microware Systems Corp.

***** ALFORD'S GOOD NEWS *****

PRE-INVENTORY CLEARANCE SALE!!

Its inventory time around here. To make things easier on ourselves, we want to sell as much of our current inventory as we can. So until stock is depleted, we are offering some FANTASTIC DEALS on selected items. Remember, these prices are ONLY good until our current stock is gone, so call or write to place your order today!

VS-1 SPEECH SYNTHESIZER.....\$120

This is the most popular unlimited speech synthesizer available for the SS-50 I/O bus! All units are completely assembled and tested, LESS the SC-01A synthesizer I.C. If you buy this unit while on sale, we will include a copy of our SC-1 text translator program at no additional charge. This gives you a \$170 value. You save \$50!! Be sure to specify 6800 FLEX, 6809 FLEX, OS-9, DOS68 or DOS69 operating system and disk size when ordering.

SP-1 SPEECH SYNTHESIZER.....\$110

Our popular unlimited speech synthesizer for the Radio Shack Color Computer is also available on this sale. All units are completely assembled and tested, LESS the SC-01A synthesizer I.C. Buy your SP-1 on this sale and we will INCLUDE our SC-1/COLOR translator program which lets your CoCo speak English using nothing more than PRINT statements! Also, we will throw in our COLOR-TREK+ program. This is our COLOR-TREK program adapted for the SP-1. English messages are continually spoken during game play! The action is non-stop. This makes a \$170+ value. You save 50+! Software is available on CASSETTE ONLY.

SC-01A SYNTHESIZER IC'S.....\$ 55

These chips are the ones used in our synthesizer boards, and are prime, new parts directly from Votrax. We have a limited number of this most popular speech synthesizer I.C. in stock for this sale. Suggested retail price is \$85. Discount houses are selling them for \$70. You save \$15 on the discount house price. We include copies of the Votrax spec sheets with these parts.

SCREDITOR-III.....\$150

All of our customers who have used both of the other major word processing packages available for 68xx systems tell us that SCREDITOR III is the best program that they have ever seen (in truth, ONE customer liked one of the others better, but you can't please EVERYONE!). As of July 15th, SCREDITOR III will cost \$175. If you have been considering a word processor, or just need a better editor than what you are using now, this is the time to buy SCREDITOR III! For more information about this fantastic program which includes over 150 pages of documentation (and another 240 pages coming soon!), see our ads in the past several issues of Micro-Journal. Be sure to specify 6800 FLEX, 6809 FLEX, OS-9, DOS68 or DOS69 operating systems and disk size when ordering.

ORDERING INFORMATION...

All sales are FINAL. No returns or refunds, NO open account orders and NO DEALER DISCOUNTS on VS-1, SP-1 or SC-01A purchases for the duration of this sale. Normal dealer discounts and open account terms will apply to SCREDITOR III purchases.

All overseas orders MUST include \$10 for shipping. All North American orders under \$100 MUST include \$5 for shipping. ALL checks MUST be drawn in U.S. dollars on accounts in U.S. banks. No foreign banks or currencies will be accepted under ANY conditions. All Virginia orders MUST include 4% state sales tax. COD orders in the U.S. ONLY, and must be shipped by UPS. All overseas orders shipped by AIR PARCEL POST. We gladly accept VISA and MASTERCARD orders.

Please allow two to three weeks for order processing. Orders paid by check may take longer. Stocks are limited. Any orders received after stocks are exhausted will be returned unprocessed.

**ALFORD AND ASSOCIATES
PO BOX 6683
RICHMOND VA 23230
PH 804-320-6722**

FLEX is a trademark of Technical Systems Consultants. OS-9 is a trademark of Microware Systems Corporation.

The Lord is our primary associate. Too often with the pressures of business today, time is not taken to acknowledge His presence, and too often we fail, due to our human weaknesses, to show His influence in our lives. We thank each of our customers for their patience and prayers for us when we do not reflect Christ in our business activities.

WE CAN EMULATE THIS CHIP BETTER THAN ITS MAKER.

Advanced Digital's Test Station Integrates and Debugs Your System Faster.

Introducing Advanced Digital's new 4009B Test Station. The test station that combines emulation and logic analysis into one system. Until now, most emulators operated in real-time by restraining the target system. Not with Advanced Digital. Our 4009B Test Station is totally transparent. Totally real-time. We meet or exceed manufacturer's timing specifications. This is particularly important when emulating interrupt intensive systems and accommodating DMA operation.

Universal operation enables you to emulate 6809 and 6809E series microprocessors without making software or hardware changes. Up to 128K of memory overlay, four hardware triggers and 2K deep of trace help you pinpoint problems faster. A line-by-line assembler and a

powerful trace disassembler are standard. They enable you to view software and make changes without returning to the development system.

MORE THAN AN EMULATOR

As a "digital engineering test station," Advanced Digital's 4009B gives you a synergistic approach to integration problems. Two microprocessors control the system. One is dedicated to emulation while the second microprocessor supervises the 32 channel logic analyzer and all system menus. This design lets you emulate and perform all system functions simultaneously.

Discover the most powerful diagnostic tool available for hardware and software integration.

Call Advanced Digital Technology for a free demonstration.



4009B TEST STATION



**Advanced
Digital
Technology**

13400 Northup Way, #27
Bellevue, WA 98005
(206) 643-2382

TEN MOST-ASKED QUESTIONS about **DYNACALC**™

THE ELECTRONIC SPREAD-SHEET FOR 6809 COMPUTERS

1. What is an electronic spread-sheet, anyway?

Business people use spread-sheets to organize columns and rows of figures. DYNACALC simulates the operation of a spread-sheet without the mess of paper and pencil. Of course, corrections and changes are a snap. Changing any entered value causes the whole spread-sheet to be re-calculated based on the new constants. This means that you can play, 'what if?' to your heart's content.

2. Is DYNACALC just for accountants, then?

Not at all. DYNACALC can be used for just about any type of job. Not only numbers, but alphanumeric messages can be handled. Engineers and other technical users will love DYNACALC's sixteen-digit math and built-in scientific functions. You can build worksheets as large as 256 columns or 256 rows. There's even a built-in sort command, so you can use DYNACALC to manage small data bases — up to 256 records.

3. What will DYNACALC do for ME?

That's a good question. Basically the answer is that DYNACALC will let your computer do just about anything you can imagine. Ask your friends who have VisiCalc™, or a similar program, just how useful an electronic spread-sheet program can be for all types of household, business, engineering, and scientific applications. Typical uses include financial planning and budgeting, sales records, bills of material, depreciation schedules, student grade records, job costing, income tax preparation, checkbook balancing, parts inventories, and payroll. But there is no limit to what YOU can do with DYNACALC.

4. Do I have to learn computer programming?

NO! DYNACALC is designed to be used by non-programmers, but even a Ph.D. in Computer Science can understand it. Even experienced programmers can get jobs done many times faster with DYNACALC, compared to conventional programming. Built-in HELP messages are provided for quick reference to operating instructions.

5. Do I have to modify my system to use DYNACALC?

Nope. DYNACALC uses any standard 6809 configuration, so you don't have to spend money on another CPU board or waste time learning another operating system.

6. Will DYNACALC read my existing data files?

You bet! DYNACALC has a beautifully simple method of reading and writing data files, so you can communicate both ways with other programs on your system, such as the Text Editor, Text Processor, Sort/Merge, STYLOGRAPH™ word processor, RMS™ data base system, or other programs written in BASIC, C, PASCAL, FORTRAN, and so on.

7. How fast is DYNACALC?

Very. Except for a few seldom-used commands, DYNACALC is memory-resident, so there is little disk I/O to slow things down. The whole data array (worksheet) is in memory, so access to any point is instantaneous. DYNACALC is 100% 6809 machine code for blistering speed.

8. Is there a version of DYNACALC for MY system?

Probably. You need a 6809 computer (32k minimum) with FLEX™, UNIFLEX™, or OS-9™ operating system. You also need a decent crt terminal, one with at least 80 characters per line, and direct cursor addressing. If your terminal isn't smart enough for DYNACALC, you probably need a new one anyway. The UNIFLEX and OS-9 versions of DYNACALC allow you to mix different brands of terminal on the same system. There's also a special version of DYNACALC for Color Computers equipped with FLEX (Frank Hogg or Data-Comp versions).

9. How much does DYNACALC cost?

The FLEX versions are just \$200 per copy; UNIFLEX version \$395; OS-9 version (works with LEVEL ONE or LEVEL TWO) \$250. Orders outside North America add \$7 per copy for postage. We encourage dealers to handle DYNACALC, since it's a product that sells instantly upon demonstration. Call or write on your company letterhead for more information.

10. Where do I order DYNACALC?

See your local DYNACALC dealer, or order directly from CSC at the address below. We accept telephone orders from 10 am to 6 pm, Monday through Friday. Call us at 314-576-5020. Your VISA or MasterCard is welcome. Please specify diskette size for FLEX or OS-9 versions. Software serial number is required for the UNIFLEX version.

Order your **DYNACALC** today!

Foreign Dealers:

Australia & Southeast Asia: order from Paris Radio Electronics, 161 Bunnerong Road (PO Box 380) Kingsford, 2032 NSW Australia. Telephone: 02-344-9111.

United Kingdom: order from Compusense, Ltd., PO Box 169, London N13 4HT. Telephone: 01-882-0681.

Scandinavia: order from Swedish Electronics hk AB, Murargatan 23-25, uppsala S-754 37 Sweden. Telephone: 18-25-30-00.

Computer Systems Center

13461 Olive Blvd.
Chesterfield, MO 63017
(314) 576-5020



UNIFLEX software prices include maintenance for the first year.

DYNACALC is a trademark of
Computer Systems Center

VisiCalc is a trademark of VisiCorp.
STYLOGRAPH is a trademark of Great Plains Computer Co.
RMS is a trademark of Washington Computer Services.
FLEX and UNIFLEX are trademarks of TSC.
OS-9 is a trademark of Microware and Motorola.

COMPARE

our EPROM PROGRAMMER with the field.

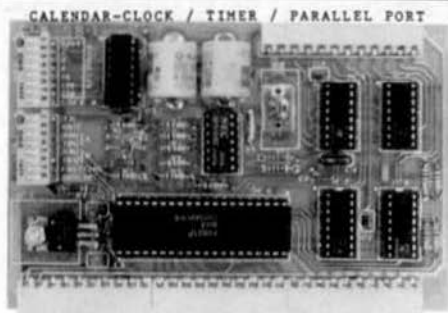
All data taken directly from manufacturer's current advertising. Software, interfaces, or personality modules may also be required at additional cost.

- Triple voltage EPROM
- Supplied in kit form

		A	B	C	D	E	F
INTERFACE	S30	PAR	PAR	SER	S30	SER	SER
INTELLIGENT	NO	NO	NO	YES	NO	YES	YES
PROGRAMS							
2704*			•				•
2508	•		•	•	•	•	•
2708*		•	•	•	•	•	•
2758	•	•	•	•	•	•	•
2516	•	•	•	•	•	•	•
2716	•	•	•	•	•	•	•
2718*	•	•	•	•	•	•	•
2532	•	•	•	•	•	•	•
2732	•	•	•	•	•	•	•
2732A	•	•	•	•	•	•	•
2584	•	•	•	•	•	•	•
2764	•	•	•	•	•	•	•
2528	•	•	•	•	•	•	•
27128	•	•	•	•	•	•	•
2616	•	•	•	•	•	•	•
68764		•	•	•	•	•	•
8748						•	•
8749						•	•
TOTAL	11	3	12	6	11	11	11
PRICE	\$125	\$45	\$169	\$289	\$375	\$489	\$575

EPROM PROGRAMMER, \$125. Versatility module for 2508, 2758, 2516, and 2716 included. Specify CPU, disk size, and operating system (PSC's PLEX or SSB's DOS) when ordering. Manual only, \$10; refundable with EPROM purchases.

UNITEK - P.O. Box 671 - Emporia, VA 23847



Calendar - Clock

CLK68-1

- Keeps date and time accurate to one second in a year
- All time functions are non-volatile
- On-board battery (included) can maintain accuracy for months
- Set of 1000, month/day/year, non-volatile (12/2/80)

Interval Timer

- For precise timing, multi-timing, etc.
- Compatible with 8080 and 8085
- 8085 runs faster with CLK68-1 than with timers such as 555 or 556
- Generates interrupt output to 8080 or 8085

Parallel I/O Port

Full featured 8 bit parallel port

- 8 bit parallel output or input 255/000 to 255/255 on the board
- Compatible with parallel port drivers to most versions of BASIC

Construction

Full featured, better version, 8 bit version

Manual -- Well documented - 36 pages

Includes a DIN connector

Assembled and tested	\$119.95	Kit	\$89.95
Goldplated bus conn	7.50	2 MHz option	2.50
Disk 5 or 8 in. SSB or Flex	OS-9 Available NOW		14.95

** 8080 is a trademark of Intel Corporation
* 8085 is a trademark of Intel Corporation

ROBERTSON ELECTRONICS
1003 Warm Sands Dr. SE
Albuquerque, NM 87123

Phone (505) 294-0025
NM residents add 4% tax
Add \$3 Shipping & Handling

'68' MICRO JOURNAL

- ★ The only ALL 6800 Computer Magazine.
- ★ More 6800 material than all the others combined:

MAGAZINE COMPARISON

(2 years)

Monthly Averages

KB	BYTE	6800 Articles	CC	DOBB'S	TOTAL PAGES
7.8	6.4	2.7	2.2	19.1 ea. mo.	

Average cost for all four each month: \$6.53
(Based on advertised 1-year subscription price)

'68' cost per month: \$2.04

That's Right! Much, Much More

for About

1/3 the Cost!

OK, PLEASE ENTER MY SUBSCRIPTION

Bill My: Master Charge ☐ — VISA ☐

Card # _____ Exp. Date _____

For ☐ 1-Year ☐ 2 Years ☐ 3 Years

Enclosed: \$ _____

Name _____

Street _____

City _____ State _____ Zip _____

My Computer Is: _____

68 Micro Journal
9900 Cassandra Smith Rd.
Hixson, TN 37343

SUBSCRIPTION RATES

USA

1 Year \$24.50, 2 Year \$42.50, 3 Year \$64.50

*FOREIGN SURFACE Add \$12.00 per Year to USA Price

*FOREIGN AIRMAIL Add \$36.00 per Year to USA Price

**CANADA & MEXICO Add \$5.50 per Year to USA Price

Cash (USA) or drawn on a USA Bank!!!



ARTISAN SYSTEMS CORP.

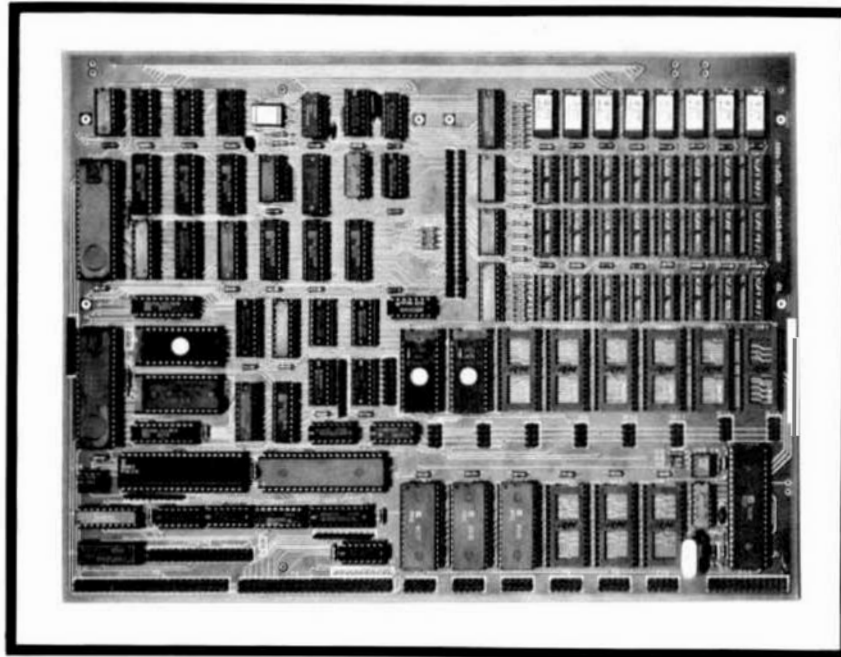
DP-09 6809 BASED SINGLE BOARD SYSTEM

2 MHZ
OPERATION

256K DRAM

6 RS-232
SERIAL PORTS

8 28-PIN
SOCKETS



DUAL 68B09E
PROCESSORS

FLOPPY
CONTROLLER

WINCHESTER
INTERFACE

4 LAYER
PC BOARD

FEATURES:

- 68B09E advanced 8/16 bit system processor with memory management hardware allows for one megabyte address space.
- 64K - 256K byte DRAM.
- 8 each 28-pin sockets for up to 128K byte EPROM. EPROMS can be 2732, 2764 or 27128. In addition 2K x 8 or 8K x 8 static RAMS may be used.
- Six RS-232 serial ports with full modem handshake. Advanced 6551A ACIAs with software baud rate select of 110 to 19.2K baud.
- 6522A interface chip provides two 16-bit timers plus two 8-bit parallel ports (unbuffered)
- 8 auto-vectored interrupts for high speed I/O handling
- 50-pin expansion socket
- Independent 68B09E subsystem for disk control 1-4 floppys 5" or 8" DS, SD, DD. SASI interface allows 5-45 MBYTE Winchesters to be connected using external controller. The subsystem uses a proprietary DMA technique for high speed operation.
- FLEX operating system is available.

PRICES

DP-09 A&T 64K, 4 Serial Ports **\$895**
FLEX for DP-09..... **\$150**
OS-9 Level One **CALL**
Forth **CALL**

ARTISAN SYSTEMS CORP.
410 CROSS STREET
WINCHESTER, MA 01890
(617) 721-2109

ALL ORDERS PREPAID. VISA & MASTERCARD ACCEPTED. 3-4 WEEKS DELIVERY. (ADD 2 WEEKS FOR PERSONAL CHECKS)
FLEX™ is a trademark of Technical Systems Consultants. OS-9™ is a trademark of Microware.

FEATURES THE
POWERFUL, THIRD
GENERATION,
MOTOROLA 6809
PROCESSOR!

THE 6809 "UNIBOARD"™ SINGLE BOARD COMPUTER KIT

PERFECT FOR COLLEGES, OEM'S, INDUSTRIAL
AND SCIENTIFIC USES!

64K RAM! DOUBLE DENSITY
FLOPPY DISK CONTROLLER!

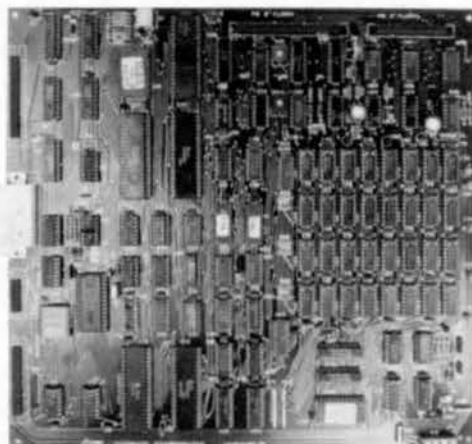
New!

BLANK PC BOARD

\$99⁹⁵

WITH PAL'S, AND
TWO EPROMS.

FOR 5-1/4 OR 8 INCH
SOURCE DISKETTE
ADD \$10.



\$399⁰⁰

COMPLETE KIT!
FULLY SOCKETED.

ALL OPTIONS ARE
STANDARD. NO
EXTRAS TO BUY!

THE COMPACTA UNIBOARD™: Through special arrangement with COMPACTA INC., we are proud to have been selected the exclusive U.S. Mfg. of their new 6809 UNIBOARD™ COMPUTER KIT. Many software professionals feel that the 6809 features probably the most powerful instruction set available today on ANY 8 bit micro. Now, at last, all of that immense computing power is available at a truly unbelievably low price.

FEATURES:

- * 64K RAM using 4116 RAMS.
- * 6809E Motorola CPU.
- * Double Density Floppy Disk Controller for either 5-1/4 or 8 inch drives. Uses WD1793.
- * On board 80 x 24 video for a low cost console. Uses 2716 Char. Gen. Programmable Formats. Uses 6845 CRT Controller.
- * ASCII keyboard parallel input interface. (6522)
- * Serial I/O (6551) for RS232C or 20 MA loop.
- * Centronics compatible parallel printer interface. (6522)
- * Buss expansion Interface with DMA channel. (6844)
- * Dual timer for real time clock application.
- * Powerful on board system monitor (2732). Features commands such as Go To, Alter, Fill, Move, Display, or Test Memory. Also Read and Write Sectors. Boot Normal, Unknown, and General Flex™.

YOUR CHOICE OF POPULAR DISK OPERATING SYSTEMS:

FLEX™ from TSC	\$149
OS9™ from Microware	\$199
Specify 5-1/4 or 8 Inch	

PC BOARD IS
DOUBLE SIDED, PLATED THRU
SOLDER MASKED, 11 x 11-1/2 IN.

Digital Research Computers
(OF TEXAS)

P.O. BOX 461565 • GARLAND, TEXAS 75046 • (214) 271-3538

TERMS: Shipments will be made approximately 3 to 6 weeks after we receive your order. VISA, MC, cash accepted. Add \$4.00 shipping. USA AND CANADA ONLY

ALL SALES ARE MADE SUBJECT TO THE TERMS OF OUR 90 DAY
LIMITED WARRANTY. A FREE COPY IS AVAILABLE UPON REQUEST.

64K SS-50 STATIC RAM

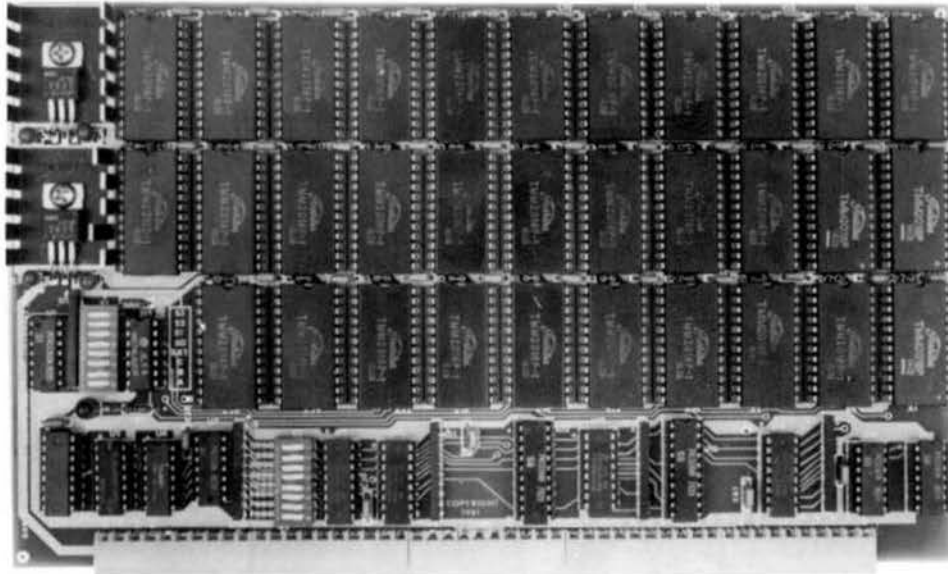
\$179⁰⁰
(48K KIT)

NEW!

NEW!

**LOW
POWER!**

**RAM
OR
EPROM!**



**BLANK PC BOARD
WITH DOCUMENTATION
\$52**

**SUPPORT ICs + CAPS - \$18.00
FULL SOCKET SET - \$15.00**

ASSEMBLED AND TESTED ADD \$50

FEATURES:

- ★ Uses new 2K x 8 (TMM 2016 or HM 6116) RAMs.
- ★ Fully supports Extended Addressing.
- ★ 64K draws only approximately 500 MA.
- ★ 200 NS RAMs are standard. (TOSHIBA makes TMM 2016s as fast as 100 NS. FOR YOUR HIGH SPEED APPLICATIONS.)
- ★ Board is configured as 3-16K blocks and 8-2K blocks (within any 64K block) for maximum flexibility.
- ★ 2716 EPROMs may be installed anywhere on Board.
- ★ Top 16K may be disabled in 2K blocks to avoid any I/O conflicts.
- ★ One Board supports both RAM and EPROM.
- ★ RAM supports 2MHZ operation at no extra charge!
- ★ Board may be partially populated in 16K increments.

56K	\$219
64K	\$249

16K STATIC RAMS?

CLOSE OUT SPECIAL
WE HAVE DROPPED OUR 32K SS-50 STATIC RAM BOARD WHICH USED 2114 LOW POWER RAMS. WE WILL SELL THE REMAINING STOCK OF BLANK PC'S WITH DATA FOR \$17.50 EA. THESE FORMERLY SOLD FOR \$50.

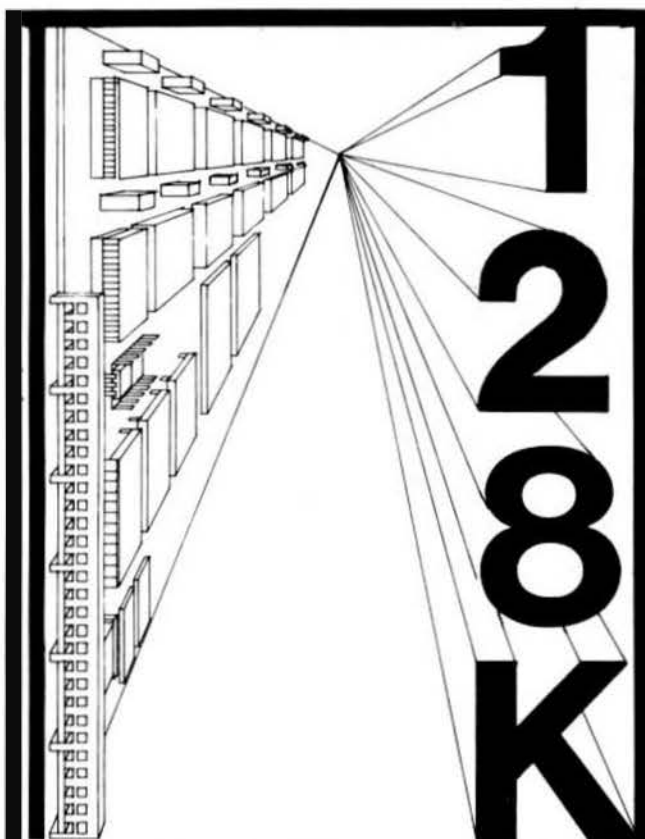
The new 2K x 8, 24 PIN, static RAMs are the next generation of high density, high speed, low power, RAMs. Pioneered by such companies as HITACHI and TOSHIBA, and soon to be second sourced by most major U.S. manufacturers, these ultra low power parts, feature 2716 compatible pin out. Thus fully interchangeable ROM/RAM boards are at last a reality, and you get BLINDING speed and LOW power thrown in for virtually nothing.

Digital Research Computers

(OF TEXAS)

P.O. BOX 401565 • GARLAND, TEXAS 75040 • (214) 271-3538

TERMS: Add \$2.00 postage. We pay balance. Order under \$15 add 75c handling. No. C.O.D. We accept Visa and MasterCard. Tex. Res. add 5% Tax. Foreign orders (except Canada) add 20% P & H. Orders over \$50. add 85c for insurance.



PRINT BUFFER or DISK EMULATOR

- Holds over 50 average-size pages
- On board firmware with many software features
- Serial or Centronics type parallel versions available on print buffer
- MC6809 on board microprocessor
- Plugs directly into SS-30 and SS-30C computers (SWTPc, GIMIX, SSB and others)
- Tested, complete with owner's manual

• No additional software necessary
 VDE-128 (Disk Emulator).....350.00
 VCM-128P:Parallel Version.... 360.00
 VCM-128S:Serial Version.....370.00
 Cable Assembly..... 30.00

NEW! EPROM PROGRAMMER FOR 2716/2732 TYPE

EPR-32.....235.00
 (complete with power supply and RS-232 interface)

Please add \$5.00 for USA orders, \$15.00 for overseas shipping/handling.

MC/VISA



Scientific Instruments

204 N. Link Lane, Alpha 9
 Fort Collins, Colorado 80524
 (303) 484-1913

ARCADE 50

POWERFUL COLOR GRAPHICS

Uses the new TMS9918A Video Display processor. High resolution 256 x 192 pixel display with 15 colors. 16K Bytes of onboard RAM does not reduce user memory. 32 graphic images can be individually moved with simple X-Y commands for smooth animation. External Video input allows subtitling. NTSC composite video output.

SOUND EFFECTS AND MUSIC

- Three AY3-8910 Programmable Sound Generators
- Nin. simultaneous voices
- Three independent noise sources
- Onboard stereo amplifier drives two 8 ohm speakers

ADDITIONAL I/O CAPABILITIES

- Eight analog inputs with 8 bit resolution
- Supports four joysticks with pushbutton switch
- Eight bit parallel I/O port
- Entire unit maps into 256 bytes of memory

FBASIC

TERMINUS DESIGN INC., in conjunction with Microware Systems Corporation, is proud to announce FBASIC an enhancement of Microware's 6800/ BASIC. Their fast compiled BASIC has been adapted for 6809 users with added video and sound features for ARCADE 50 users. FBASIC is a true compiler that produces optimized machine language modules which are ROMable and require no Run-Time package. FBASIC requires less memory overhead and runs hundreds of times faster than BASIC interpreters. It supports standard BASIC instruction including String functions. Disk I/O and fast integer arithmetic with multiple-precision capability. Graphics verbs and functions fully support the Arcade 50.

ARCADE 50 assemble and tested	\$325.00
Video and Audio connector set	15.00
4 Joystick connector set	15.00
2 Radio Shack Joysticks	24.00
Gold Molex connectors	12.00
A/BASIC for 6800	110.00
FBASIC for 6809	110.00
FBASIC (with ARCADE 50)	75.00
ARCADE 50 R/B	375.00
LABVIDEO I/Motorola EXORbus	375.00
NEW MV09 6809 Processor Board	225.00
256K Dynamic Memory Board	795.00
256K Dynamic Memory Board (w/64K)	395.00
64K Dynamic Memory Board	295.00

TERMINUS DESIGN INC
 16 SCARBROUGH ROAD
 ELLENWOOD, GA 30049

TERMINUS DESIGN, INC. TEL: (404) 474-4866

68 MICRO JOURNAL PROGRAMS on DISK

Disk #1: FILESORT, MINICAT, MINICOPY, MINIFMS, **LIFETIME, **POETRY, **FOODLIST, **DIET.
 Disk #2: DISKEDIT w/ inst. & files, PRIME, **PRMO, **SNOOPY, **FOOTBALL, **HEXPWN, **LIFETIME.
 Disk #3: CBUG09, SEC1, SEC2, FIND, TABLE2, INTENT, DISK-EXP, **DISKSAVE.
 Disk #4: MAILING PROGRAM, *FINDDAT, *CHANGE, *TESTDISK.
 Disk #5: *DISKFIX 1, *DISKFIX 2, **LETTER, **LOVESIGN, **BLACKJAK, **BOWLING.
 Disk #6: **PURCHASE ORDER, INDEX (Disk file Indx).
 Disk #7: Linking Loader & RLOAD, Harkness
 Disk #8: CRTSET, Lanpher (May '82)
 Disk #9: DATECOPY, DISKFIX9 (Aug '82)

NOTE: All are as published or received by 68 Micro Journal, some have fixes and patches.

This is a reader service only! No Warranty is offered or implied, they are as received and are for reader convenience ONLY. Also 6800 and 6809 programs are mixed, as each is fairly simple (mostly) to convert to the other.

PRICE: 8" Disk \$19.95 - 5" Disk \$17.95

68 MICRO JOURNAL
 POB 794
 Hixson, TN 37343
 615-842-4600

* Indicates 6800, ** Indicates BASIC SWTPC or TSC - 6809 no Indicator.

MASTER CARD - VISA accepted - Foreign add sufficient postage surface or air!!

SS-50 SOFTWARE

FROM THE WORLD'S LARGEST SUPPLIER OF SOFTWARE
FOR FLEX AND OS-9!!!

A/BASIC COMPILER - generates pure, fast efficient 6809 machine code from easy to write BASIC source programs. For 6809 OS-9 or FLEX - \$150.00

DYNASTAR - Screen Editor - A powerful menu-driven screen editor equally suited to the tasks of program preparation and document processing. For 6809 OS-9 or FLEX - \$149.95

DYNAPORM - Text Formatter - Provides all the standard features such as pagination, headers and footers with page numbers, single space, double space, bold face, double strike and more. Purchase with DYNASTAR for complete word processing system. For 6809 OS-9 or FLEX - \$149.95

DYNASOFT PASCAL - A fast integer only P-Code compiler ideal for ROMable applications. Powerful enough for the DYNASTAR word processor. Written for 6809 OS-9 - \$69.95 or FLEX - \$59.95 (Run-time source is additional \$30.00)

DYNA "C" - A very fast middle "C" from the people who brought you Dynasoft Pascal. FLEX - \$99.95 OS-9 - \$124.95

HELP - A data retrieve utility designed to save you time digging through manuals looking for info about computer language commands and statements. Written in assembler for 6809 FLEX - \$29.95

AUTOTASK - Consists of a set of memory resident commands that are aids in the design of user oriented applications programs. Written in assembler for 6809 FLEX - \$79.95

w/MCOMMAND - a utility for converting disk resident commands to memory resident commands-\$50.00 (Free with purchase of Autotask)

YDISK - Provides a way for FLEX users to take advantage of a large memory array. Permits the user to treat extended memory as an additional disk drive. Written for 6809 FLEX - \$100.00

READTEST - A must for all writers and instructors, it reads your text and tells you how well it was written. In assembler for 6809 FLEX - \$54.95

CRASMB - Will cross assemble source code into object code. The source or binary available for the following: 6800, 6801, 6809, 6502, 1802, Z80, and Z8. OS-9 version includes 6809 binary. (Use 6809 system to develop software for 6803 Micro Color Computer) OS-9 and FLEX \$200.00. CPM's additional.

OSM - A fast and versatile macro assembler with ability to define macros with substantial parameters, conditional assembly directives and ability to change value of a label or symbol. Create OS-9 binary files in FLEX and vice versa! For OS-9 or FLEX - \$125.00

ED/ASM - ED has all the features of TSC's EDITOR with the addition of screen type editing, MACRO capability, and a math package. ASM is compatible with TSC's ASSEMBLER, but with more powerful MACROS and conditionals than TSC's. For 6809 FLEX - \$69.95

DYNASPELL - The most versatile 68XX spelling checker available. Fast and easy to use. In assembler for 6809 OS-9 or FLEX - \$199.00

PL-9 - A co-resident EDITOR COMPILER/DEBUGGER written by Graham Trott. A single pass compiler that produces position independent machine code output. For 6809 FLEX - \$198.00 (Exclusive US distributors).

MACE - A co-resident EDITOR ASSEMBLER written by Graham Trott. Includes XMACE, a co-resident 6800/1/3 EDITOR/CROSS ASSEMBLER. For 6809 FLEX - \$98.00

PHL EXTENDED USE UTILITIES - A set of 12 utilities that add the final touch to your utilities for FLEX. Written in assembler for 6809 FLEX - \$49.95

TOOLKIT #1 - Add EDIT capabilities to TSC's BASIC, along with DECOMPIL and cross reference. For 6809 FLEX - \$49.95

TOOLKIT #2 - A package of utilities and programs developed to extend the capabilities of the FLEX operating system. Written in assembler for 6809 FLEX - \$49.95

FHI FRANK HOGG LABORATORY

THE REGENCY TOWER • SUITE 215 • 770 JAMES ST. • SYRACUSE, NY 13203
PHONE (315) 474-7856 • TELEX 646740

6809 Word Processing System

*stylograph*TM

STYLOGRAPH 2.0

The "User Friendly" word processing system. Fewer key strokes by the operator make it easier to learn.

OS9, FLEX \$295 UnifLEX \$395
COLOR COMPUTER FLEX \$195

SPELLING CHECKER

Checks all words against an internal user-expandable dictionary of over 42,000 words.

OS9, FLEX \$145 UnifLEX \$195

MAIL MERGE

Inserts names and addresses into form letters and mailing lists. Appends files at print out time. Handles files longer than memory.

OS9, FLEX \$125 UnifLEX \$175

Inquire about our other software

- Business Programs - G/L, A/R, A/P
- Data Base Management System
- Assemblers

Also, Daisy Wheel Printers \$599.

Great Plains Computer Company Inc.
P.O. Box 916
Idaho Falls, Idaho 83401
(208) 529-3210

Flex and Uniflex are trademarks of Technical Systems Consultants, Inc.
OS9 is a trademark of Microware.

UNEQUAL[®] in the 6809 ARENA

AM-100[™] ADVANCED WINCHESTER SubSystems for FLEX[™] and OS-9[™] COMPLETE[™] AM-100 starting as low as \$1899.-

• Intelligent microprocessor based controller • extensive error diagnostics • standard SASI interface • host interface assures efficient data transfer via hardware implemented handshaking • powerful software drivers • complete floppy/winchester integration, logical division of winchester up to 4 logical drives • quick & easy installation: no modification required to user's hardware or software • GET RID OF ANNOYING DISK ERRORS • NO MORE SLOW FLOPPY OPERATIONS!

*UNEQUAL QUALITY • UNEQUAL PRICES • UNEQUAL SUPPORT
TO OUR USERS AND DEALERS • CALL AND FIND OUT

complete[™] subsystems

5MB FIXED—	STD. 5" —	\$1899.-
10MB FIXED—	STD. 5" —	\$2099.-
5MB FIXED—	MINI 5" —	\$2315.-
5MB REMOVABLE—	MINI 5" —	\$2395.-

**COMPLETE SUBSYSTEM — Drivetail • Controller • Power Supply • Enclosure • All Cables • Host Interface • Software Drivers • NO HIDDEN EXTRAS!!!

options

SS-50 HOST INTERFACE (incl. SOFTWARE DRIVERS)	\$325.-
SET OF WINCHESTER SOFTWARE UTILITIES	\$49.-
NOW TO WINCHESTERIZE — A COMPLETE MANUAL	\$18.-
shipping & handling 2.00 (US, Canada) • 4.00 (overseas)	

TERMS: • VISA • MONEY ORDER •

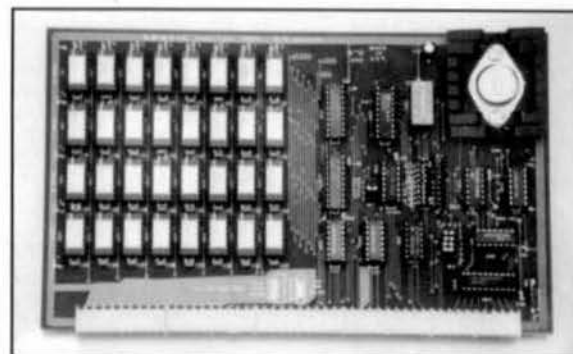


interfacing technologies corp.

P.O. BOX 578 Snowdon (514) 737-8787

4890 Bourret Ave. Montreal Quebec, H3K 3T7

Interfacing Technologies Corp. is a subsidiary of Technical Systems Consultants Inc. © 1984 Microware Inc.



QMM2 \$795 256K DYNAMIC MEMORY

Works in SS50-C 6809 systems by SWTPC, SSR, GIMIX, and others. Runs at 1 or 2 Mhz with on board refresh. Comes assembled, tested, burned in and warranted for 1 year.

Professional quality board with socketed components and gold connectors.

Also available with 64K, 128K, or 192K.

Delivery: Stock—2 weeks.

Terms: Prepaid, C.O.D., VISA, MASTERCARD.

D.P. Johnson (503) 244-8152
7855 S.W. Cedarcrest St., Portland, OR 97223

SUPER CPU

... only from
LSI



Yes, it's SUPER-CPU, advanced, new 32 bit processor with power and ability far beyond those of normal CPU's. SUPER-CPU, who can change the course of mighty SS-50 processing, break 68XX benchmarks with its bare MC68008 CPU. And who, disguised as LSI's S68K/08-CPU SS-50c compatible card, fights a never ending battle for speed, power and the SS-50c user.

We're mighty proud of our new processor card. We're giving you the ability to go 68000 without major changes to your system. Our new CPU gives you these advanced features:

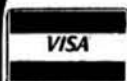
- Dynamic partitioning memory management unit with bound check register.
- On-board timer for multi-user/multi-tasking applications.
- On-board boot-strap EPROM and Monitor EPROM space.
- Vectored priority interrupt generator.
- On-board wait state generator.
- User selectable bus options that includes a new higher bandwidth bus mode.
- And many more...

S68K/08-CPU.....\$799.95

CP/M-68K.....\$350.00

(contact factory for disk controllers supported)

CP/M-68K is a registered trademark of Digital Research, Inc.



N.Y. residents add sales tax.

LSI Enterprises Ltd.

PO Box 1227

Woodhaven, NY 11421

(212) 631-9242

OS9 APPLICATION SOFTWARE

ACCOUNTS
PAYABLE

\$299

GENERAL
LEDGER
with
CASH

JOURNAL

\$399

PAYROLL

\$499

ACCOUNTS
RECEIVABLE

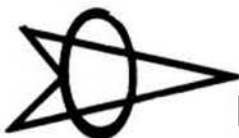
\$299

SMALL
BUSINESS
INVENTORY

\$299

COMPLETE DOCUMENTATION \$19.95

OS9 & BASIC 09 ARE TRADEMARK OF
MICROWARE, INC. & MOTOROLA CORP.



SPECIALTY ELECTRONICS

(405) 233-5564

2110 W. WILLOW - ENID, OK 73701

ANDERSON COMPUTER CONSULTANTS & Associates

Ron Anderson, respected author and columnist for 68 MICRO JOURNAL announces the Anderson Computer Consultants & Associates, a consulting firm dealing primarily in 68XX(X) software design. Our wide experience in designing 6809 based control systems for machine tools is now available on a consultation basis.

Our experience includes programming machine control functions, signal analysis, multi-axis servo control (CNC) and general software design and development. We have extensive experience in instrumentation and analysis of specialized software. We support all popular languages pertaining to the 6809 and other 68XX(X) processors.

If you are a manufacturer of a control or measuring package that you believe could benefit from efficient software, write or call Ron Anderson. The fact that any calculation you can do with pencil and paper, can be done much better with a microcomputer. We will be happy to review your problem and offer a modern, state-of-the-art microcomputer solution. We can do the entire job or work with your software or hardware engineers.

Anderson Computer Consultants & Associates
3540 Sturbridge Court
Ann Arbor, MI 48105

DYNASHARE

NOW SHIPPING

SOUTH EAST MEDIA is now shipping from stock DYNASHARE — the multi-user, multi-tasking capability of DYNASHARE allowing FLEX™ users the advantages of more sophisticated and time saving computer usage without having to buy or learn a new language or operating syntax vocabulary!

DYNASHARE is the painless method! Use your existing FLEX™ computer and simply add 64K of RAM for each user. Fact is you still use FLEX™ just like you always have! PLEASE NOTE THE SPECIFICATIONS SHOWN HERE.

\$200.00

SOUTH EAST MEDIA — CFI
5900 Cassandra Smith
Hixson, TN 37343
INFO: (615) 842-4600
ORDERING: 1-800-338-6800

FLEX is a trademark of Technical Systems Consultants, Inc. DYNASHARE & DYNACALC are trademarks of Computer Systems Center.

SOUTH EAST MEDIA is a division of the Computer Publishing Corporation family of 100% 68XX support facilities.

DYNASHARE, as its name implies, allows true "time-sharing" operation under the popular FLEX operating system. In addition to multi-user operation, DYNASHARE also allows each user to run two simultaneous jobs (multi-tasking).

The initial release of DYNASHARE is for SxTPC 3/09 computers, but versions will also be available for other popular extended-memory systems, such as HELIX and GIMIK. A minimum of 128K of RAM will be required in all versions.

DYNASHARE requires 64K of RAM for each active task; thus a 256K system could allow foreground-background operation on two terminals, or foreground-only operation on four terminals. Additional memory is supported to 1024K, depending on system hardware.

DYNASHARE's multi-tasking feature may be used even on single-user systems. For example, while in EDIT, you can list another file or examine a directory. Or you might look up an item in a data base even while a sort is in progress!

DYNASHARE also provides some fringe benefits that will be greatly appreciated by FLEX users, including type-ahead, command line editing, and instant response to "escape".

DYNASHARE is not intended as competition to UNIFLEX. It does not improve on the speed of FLEX, and does not offer password protection or other niceties of a full-blown multi-user system. What DYNASHARE does do is give FLEX users a low-cost way to use existing software in a multi-user, multi-tasking environment. This means that you can continue to use your existing FLEX versions of BASIC, BASIC, editors, assemblers, disassemblers, sort/merge packages, word processors, compilers, DYNACALC spreadsheet package, and so on.

DYNASHARE is in customer-acceptance testing now. It will be available for sale on May 1, 1983. Price is \$200 per copy; dealer and OEM inquiries invited.

Call 1-800-338-6800 For Ordering Only!



P.O. Box 794

Hixson, Tn. 37343

(615) 842-4601



SOFTWARE For THE HARDCORE

** FORTH PROGRAMMING TOOLS from the 68XX&X **
** FORTH specialists — get the best!! **

NOW AVAILABLE — A variety of rom and disk FORTH systems to run on and/or do TARGET COMPILATION for

6800, 6301/6801, 6809, 68000, 8080, Z80

Write or call for information on a special system to fit your requirement.

Standard systems available for these hardware—

EPSON HX-20 rom system and target compiler
6809 rom systems for SS-50, EXORCISER, STD, ETC.
COLOR COMPUTER
6800/6809 FLEX or EXORCISER disk systems.
68000 rom based systems
68000 CP/M-68K disk systems, MODEL 11/12/16

tFORTH is a refined version of FORTH Interest Group standard FORTH, faster than FIG-FORTH. FORTH is both a compiler and an interpreter. It executes orders of magnitudes faster than interpretive BASIC. MORE IMPORTANT, CODE DEVELOPMENT AND TESTING is much, much faster than compiled languages such as PASCAL and C. If Software DEVELOPMENT COSTS are an important concern for you, you need FORTH!

firmFORTH™ is for the programmer who needs to squeeze the most into roms. It is a professional programmer's tool for compact rommable code for controller applications.

™ tFORTH and firmFORTH are trademarks of Talbot Microsystems.
™ FLEX is a trademark of Technical Systems Consultants, Inc.
™ CP/M-68K is trademark of Digital Research, Inc.

tFORTH™
from TALBOT MICROSYSTEMS
NEW SYSTEMS FOR
6301/6801, 6809, and 68000

---> IFORTH SYSTEMS <---

For all FLEX systems: GIMI X, SWTP, SSB, or EXORCISER Specify 5 or 8 inch diskette, hardware type, and 6800 or 6809.

** tFORTH — extended fig FORTH (1 disk) \$100 (\$15)
with fig line editor.

** tFORTH+ — more! (3 5" or 2 8" disks) \$250 (\$25)
adds screen editor, assembler, extended data types, utilities, games, and debugging aids.

** TRS-80 COLORFORTH — available from The Micro Works
firm FORTH — 6809 only. \$350 (\$10)

For target compilations to rommable code.
Automatically deletes unused code. Includes HOST system source and target nucleus source. No royalty on targets. Requires but does not include tFORTH+.

** FORTH PROGRAMMING AIDS — alabore decompiler \$150

** tFORTH for HX-20, in 16K roms for expansion unit or replace BASIC \$170

** tFORTH/68K for CP/M-68K 8" disk system \$290
Makes Model 16 a super software development system.

** Nautilus Systems Cross Compiler
— Requires: tFORTH + HOST + at least one TARGET:
— HOST system code (6809 or 68000) \$200
— TARGET source code: 6800-\$200, 6301/6801—\$200
same plus HX-20 extensions— \$300
6809—\$300, 8080/Z80—\$200, 68000—\$350

Manuals available separately — price in ().
Add \$6/system for shipping, \$15 for foreign air.

TALBOT MICROSYSTEMS 1927 Curtis Ave., Redondo Beach, CA 90278 (213) 376 9941

THOMAS INSTRUMENTATION

HARDWARE

ASSEM & TEST

BARE CARD

SOFTWARE

CPU-5	\$299.00	\$49.00	VDISK
S-R/R w/o memory chips	\$120.00	\$49.00	Use extended memory
S-R/R with 48K	\$399.00	\$49.00	as a fast disk drive
SP-1	\$195.00	\$49.00	6809 source & obj \$149.00
6802 SUPER CPU	\$235.00	\$59.00	6809 object \$ 99.00
VIDEO RAM	\$195.00	\$49.00	
PARALLEL I/O	\$139.00	\$49.00	OUTSIDE MODEM PROGRAM
SS-50/50C EXTENDER	\$ 35.00	NA	includes source
SS-30 EXTENDER	\$ 25.00	NA	UniFLEX™ \$100.00
SS-50 WIRE WRAP	NA	\$39.00	FLEX™ 6800 or 6809 \$ 50.00
SS-30 WIRE WRAP	NA	\$20.00	
SS-30 BACKPLANE 8POS.	NA	\$39.00	CROSS ASSEMBLER
SS-50 BACKPLANE 4,8,12, & 16 @ \$5.00/SLOT	NA	NA	For 6800, 6801, 6805
			Runs on 6809 FLEX™
			\$150.00

COMPONENTS

GOLD MALE MOLEX \$1.60	GOLD FEMALE MOLEX \$1.60
TIN MALE MOLEX \$.40	TIN FEMALE MOLEX \$.50
146805E2P	\$20.00

THOMAS INSTRUMENTATION
168 EIGHTH STREET
AVALON, N.J. 08202
(609) 967-4280

MASTERCARD, VISA, AND C.O.D. ACCEPTED. PLEASE ALLOW 3 WEEKS
FOR CHECKS TO CLEAR. CONT. USA ADD \$3.00 SHIPPING. CANADA
\$6.00. FOREIGN \$12.00. NJ RESIDENTS ADD 6% SALES TAX
FLEX and UniFLEX ARE TRADEMARKS OF TECHNICAL SYSTEMS CONSULTANTS

HARDWARE DESIGNERS

PROGRAMMERS

Cut Your Design Time In Half!

Our TSL 6809/6802 MPU Emulator Allows You To Do Just That!

Tired of troubleshooting 6809/6802 based systems with an oscilloscope alone? Is the continual EPROM blowing during software development slowing you down? If your answer to either question is "yes," TSL's 6809/6802 MPU Emulator could be the time-saver you have been waiting for. Our Emulator requires no hardware modifications to your SWTPC computer.

TSL's 6809/6802 MPU Emulator for Southwest Technical Computers...

- Examines external hardware addresses from the keyboard of the SWTPC computer (using the TSL's EM-BUG monitor).
- Executes in Southwest RAM development software which addresses external hardware.
- Sets hardware breakpoint for memory location: read attempt, write attempt, and access attempt.
- Sets up to 10 software breakpoints (SWI type).

System Requirements:

SWTPC computer with 20-bit
addressing
Fully decoded memory boards
Flex operating system

Emulator package contains:

One SS-50 Emulator board
EM-BUG monitor (in ROM)
EM-SWTPC configuration program
on disk
All interconnecting cables
User's manual

**INTRODUCTORY
OFFER: \$695⁰⁰**

TERMS: VISA, MasterCard, Check or money order. Mo. residents add \$39.09
tax. Foreign countries pay in U.S. dollars. Normal Delivery: Stock to
2 weeks. Shipped U.P.S. in continental U.S.A.
Best way to foreign countries.



TSL Division
Teknar, Inc.
306 Axminster Dr.
Fenton, MO 63026
314/343-9140

MACROPLEX Software

175 Fifth Avenue, Suite 3011, New York, NY 10010

At Last... Pipelines Grow Up!
Into a powerful development and production tool

Presenting... **THE CONDUIT**
 Enhancement Software for OS9

Have you ever wanted to—

- ★ Speed up compilers and assemblers with fast disk store-down?
- ★ Redirect terminal I/O when you need it—without calling Shell?
- ★ Interconnect any compatible paths in concurrent programs?
- ★ Intercept program I/O requests to implement them your way?

THE CONDUIT can do it!

OS9 Level 1 support consists of 1.2Kb file manager, pseudo-device descriptor and driver, utilities in source and object, all on 5.25 or 8 in. disk. 50 page manual contains usage guide, coding examples, installation instructions, case study.

(Version for OS9 Level 2 available January 1984.)

Specify disk size when ordering.
 Manual \$8... Disk \$70... Both together \$75

New York State: add 7% sales tax (NYC 8.25%).
 Outside U.S., Can., Mex. add air parcel postage for 0.25 kgm.

OS9™ Microware Systems Corp. & Motorola Inc.

ACORN 
 COMPUTER SYSTEMS 88-50C

Color Computer

MONOLINK

- ★ for 18 MHz MONITOR
- ★ NO SOLDERING (A+T)
- ★ LED POWER INDICATOR
- ★ SIMUL. MONITOR + TV
- ★ with CO-AX CABLE

KIT \$1500
A+T \$2000
TAXAN 18MHz GREEN 12" MONITOR \$149.95

Write for FREE Catalog :
 ADD \$3.00 SH PER ORDER
 WIS. ADD 5% SALES TAX



11931 W. Bluemound Road
 MILWAUKEE, WIS. 53226
 (414) 257-0300

'68' MICRO JOURNAL ADVERTISERS INDEX

'68' MICRO JOURNAL	60,64
AAA CHICAGO COMPUTER CENTER	8
ACKERMAN DIGITAL SYSTEMS INC.	56
ACORN COMPUTER SYSTEMS	70
ADVANCED DIGITAL TECHNOLOGY	58
ALFORD & ASSOCIATES	57
ANDERSON COMPUTER CONSULTANTS	67
ARTISAN SYSTEMS CORP.	61
CHIRATECH SCIENTIFIC INSTRUMENTS	64
CLEARBROOK SOFTWARE GROUP INC.	71
COLOR MICRO JOURNAL	9
COMPUTER PUBLISHING INC.	5
COMPUTER SYSTEMS CENTER	47,59,68
COMPUTER SYSTEMS CONSULTANTS, INC.	56
D.P. JOHNSON	66
DATA SYSTEMS "68"	71
DATA-COMP	1BC
DIGITAL RESEARCH COMPUTERS	62,63
FRANK HOGG LABORATORY, INC.	65
GIMIX, INC.	3,72
GREAT PLAINS COMPUTER CO.	66
HAZELWOOD COMPUTER SYSTEMS	0BC
INTERFACING TECHNOLOGIES, CORP.	66
INTROL CORP.	48
JOTO ASSOCIATES	71
LSI ENTERPRISES LTD.	67
MACROPLEX SOFTWARE	70
MICROWARE SYSTEMS CORP.	4
OMEGASOFT	35
ROBERTSON ELECTRONICS	60
SMOKE SIGNAL BROADCASTING	7
SOUTH EAST MEDIA	50,51,52,53,54,55
SOUTHWEST TECHNICAL PRODUCTS CORP. ...	1FC,36,37
SPECIALTY ELECTRONICS, INC.	67
STAR-KITS	48
TALBOT MICROSYSTEMS	68
TECHNICAL SYSTEMS CONSULTANTS, INC.	1
TEKNAR, INC.	69
TERMINUS DESIGN, INC.	64
THOMAS INSTRUMENTATION	69
UNITEX	60
UNIVERSAL DATA RESEARCH, INC.	6
WINDRUSH MICRO SYSTEMS LIMITED	49

This Index is provided as a reader service. The publisher does not assume any liability for omissions or errors.

DATA SYSTEMS 68

CHECK OUT THE NEW PRICES ON THE BEST BARE BOARDS AVAILABLE!!
(THEN TRY SOME)

8" DOUBLE DENSITY DISK CONTROLLER - \$50⁰⁰
5 1/4" DOUBLE DENSITY DISK CONTROLLER - \$50⁰⁰

DMA INTERFACE - \$32⁰⁰

64K DYNAMIC RAM BOARD - \$50⁰⁰

6809 CPU BOARD - \$45⁰⁰

MOTHER BOARD - \$65⁰⁰

6845 VIDEO DISPLAY BOARD - \$45⁰⁰

6847 VIDEO GRAPHICS BOARD - \$45⁰⁰

DUAL SERIAL I/O BOARD - \$25⁰⁰ MULTIPLE I/O BOARD - \$40⁰⁰
MODEM BOARD - \$30⁰⁰ 30 & 50 PIN EXTENDER BOARDS - \$25⁰⁰ ea.

- All Boards for the SS-50 Buss
- All Solder Masked Both Sides
- All Silk Screened Nomenclature

- Full Documentation Included
- Visa & MasterCard Accepted
- Add \$2⁰⁰ for C.O.D.

- Add \$4⁰⁰ for U.S. Shipping
- Add \$5⁰⁰ for Canadian Shipping
- Add \$10⁰⁰ for Overseas Shipping

Data Systems "68"
2316 Diversified Way
Orlando, Florida 32804



(305) 425-6800



Data Systems "68"
2316 Diversified Way
Orlando, Florida 32804

Florida residents add 5% sales tax. Prices effective February 1, 1983



OS-9 SOFTWARE

Disk Utilities Package (CSG-DUP) \$49
Four utilities designed to save you time (and typing). **DDir** — display a directory listing of one or more directories or of an entire disk. **DCopy** — copy files in one or more directories or an entire disk with or without prompting. **DDel** — Delete files in one or more directories or an entire disk. **DAttr** — re-attribute files or change owner I.D. for files in one or more directories or an entire disk. All utilities allow wild card file specification (partial filename, attribute and own I.D.) and have the ability to use AND, OR and NOT logic for file specification relations.

DEdit 909 (CSG-DEDIT) — A screen oriented disk editor. DEdit allows the user to view and modify any readable sector on an OS-9 format disk. A terminal with clear screen and cursor addressing functions is required. Data is displayed in hexadecimal, ASCII and binary formats. Modifications to a sector are made by moving the cursor to the byte(s) to be changed, typing the correct data and issuing an Update command. (Written in machine language.)

BT9 \$59 (CSG-BT9) — A binary tree filing program. BT9 makes it easy to write BASIC09 programs in which files must be maintained in sorted order. Three types of files are supported: Index File, Indexed Data File and Non-indexed Data File. BT9 also allows for maintaining several indexes for one file. All sorting is performed on disk leaving more memory free for other programs. BT9 is written in machine language for fast operation (typical search time under two seconds for a file of 2000 records).

SPECIAL OFFER — Order 2 CSG products and deduct 10%. Order all 3 products and deduct 15%. Any orders received before 83/09/30 will receive CrashCopy. CrashCopy attempts to copy files from a disk on which the disk allocation map sector is damaged. Call (604) 853-9118 for information. Specify 8 inch or 5 inch format.

Send Certified Cheque or Money Order to: Clearbrook Software Group, Box 8000, Suite 499, Sumas, WA 98295, Canada. Write: Box 8000, Suite 499, Abbotsford, B.C. V2S 1K3

FIRST CLASS

INTELLICOM

INTELLigent COMUnications Program

INTELLICOM provides you with the capability of intelligent computer to computer communications from both a terminal emulation and file transfer standpoint. INTELLICOM supports several file transfer protocols that facilitate the transfer of both binary and ASCII data. Since INTELLICOM is menu drive, it is a breeze to use and understand. With INTELLICOM you will be able to communicate with the various data and timesharing services such as The Source and CompuServe. Additionally, since INTELLICOM supports the protocol used by virtually all remote CP/M systems around the world, all users can immediately begin to take advantage of the wealth of public domain software available on these systems. Current, or potential, users of CompuServe can transmit and receive both binary and ASCII data with full error detection and recovery. The checksum protocols allow for the verifications of data blocks transferred (assuming appropriate support on the host end). This feature will be of great value in those applications where data integrity is paramount. INTELLICOM's documentation includes a detailed description of all protocols used along with machine readable examples of host pseudo code. A telephone directory and dialing facility compatible with Hayes Smartmodems is supported.

Price: OS9 or FLEX Version \$99.95, add \$5.00 C.O.D.
JOTO ASSOCIATES, 104 Lepage Dr., Southington, CT 06489

203-621-8070

TO REALIZE THE FULL POWER & PERFORMANCE OF THE 6809, LOOK TO GIMIX.

GIMIX OFFERS YOU A VARIETY OF SS50 BUS COMPONENTS AND SYSTEMS.

OS-9 GIMX III

The GIMIX 6809 CPU III and OS-9 GIMIX III... A Multi-user, Multi-tasking package for the ultimate in System Performance plus protection of the system and other users from crashes caused by errors in individual users programs.

#01 (CPU & Software) **NEW!** \$1698.01

INTELLIGENT I/O PROCESSOR BOARDS increase system throughput by reducing interrupts to the host, buffering data transfers, and data preprocessing. Prices include on board firmware. Requires system drivers.

#11 3 port RS232 Serial (SS30) **NEW!** \$498.11

#12 4 port Parallel (SS50) \$538.12

OS-9 GIMIX III drivers... (Included when purchased with GIMIX III package) \$200.00

OS-9 Level 2 users - contact GIMIX for system requirements and availability.

192K GIMIX III 670 SYSTEMS: All include GIMIX 6809 CPU III and OS-9 GIMIX III (#01); a #11 3 port Intelligent serial I/O & cables; #19 Classy Chassis; 192KB Static RAM; #08 DMA controller; all necessary cables, power regulators, and filter plates. The OS-9 Editor, Assembler, Debugger, BASIC-09, and RUN9 are included.

#70 with dual 40 track DSDD drives **NEW!** \$5998.79

#71 with dual 80 track DSDD drives \$6298.79

#72 with #88 8" Dual Drive Disk System \$7998.79

#73 with #90 19MB Winchester subsystem & one 80 track DSDD drive \$8998.79

UnifLEX for the GIMIX 6809 CPU III and Intelligent I/O boards is in development.

OS-9 GIMIX I; OS-9 GIMIX II; FLEX; and UnifLEX

The #05 GIMIX 6809 PLUS CPU board \$578.05
Options: GIMIX DAT \$35.00 SWTPC DAT \$15.00
9511A \$312.00 9512 \$265.00

#09 64KB GHOST SYSTEM includes: #05 CPU; #19 Classy Chassis; 64KB static RAM; a #43 2 port serial card & cables; #08 DMA Controller; all necessary cables, power regulators, and filter plates; GIMIXBUG monitor; FLEX; and OS-9 GIMIX I. You can software select either FLEX or OS-9. The OS-9 Editor, Assembler, Debugger, BASIC-09, and RUN9 are also included.

#49 with dual 40 track DSDD drives \$4398.49

#49 with dual 80 track DSDD drives \$4698.49

#49 with #88 8" Dual Drive Disk System \$5898.49

#49 with #90 1 MB Winchester subsystems & one 80 track DSDD drive \$7398.49

#09 128KB SYSTEM includes: #05 CPU/DAT; #19 Classy Chassis; 128KB of static RAM; a #43 2 port serial card & cables; #08 DMA Controller; all necessary cables, power regulators, and filter plates; GIMIXBUG monitor; FLEX; and OS-9 GIMIX II. You can software select either FLEX or OS-9. The OS-9 Editor, Assembler, Debugger, BASIC-09, and RUN9, and GIMIX-VOISK for FLEX are included.

#39 with dual 40 track DSDD drives \$4998.39

#39 with dual 80 track DSDD drives \$5298.39

#39 with #88 8" Dual Drive Disk System \$6898.39

#39 with #90 19MB Winchester subsystem & one 80 track DSDD drive \$7998.39

UnifLEX, available at extra cost, requires 8" or Winchester drives. A signed license agreement with TSC is required before shipment.

You can add to any GIMIX system RAM, I/Os and other options, or substitute non-volatile RAM. GIMIX will customize to your needs.

COMING SOON: Contact GIMIX for price and availability on 40MB and 72MB Winchester (5 1/4") drives, removable pack Winchesters, 256KB static RAM boards.

All GIMIX systems are guaranteed for 2MHz operation. GIMIX systems include documentation for all boards and software in a GIMIX binder. ALL DRIVES ARE 100% TESTED AND ALIGNED BY GIMIX.

ALL BOARDS AND SYSTEMS ARE ASSEMBLED, BURNED-IN, AND TESTED. GOLD-PLATED BUS CONNECTORS ARE USED.

TO ORDER BY MAIL: SEND CHECK OR MONEY ORDER OR USE YOUR VISA OR MASTER CHARGE. Please allow 3 weeks for personal checks to clear. U.S. orders add \$5 handling if order is under \$200.00. Foreign orders add \$10 handling if order is under \$200.00. Foreign orders over \$200.00 will be shipped via Emery Air Freight COLLECT, and we will charge no handling. All orders must be prepaid in U.S. funds. Please note that foreign checks have been taking about 8 weeks for collection so we would advise wiring money, or checks drawn on a bank account in the U.S. Our bank is the Continental Illinois National Bank of Chicago, 231 S. LaSalle Street, Chicago, IL 60693, account #73-32033. Visa or Master Charge also accepted.

EXPORT MODELS: ADD \$30 FOR 50HZ POWER SUPPLIES.

GIMIX Inc. reserves the right to change pricing, terms, and product specifications at any time without further notice.

ALL PRICES ARE F.O.B. CHICAGO

Choose from GIMIX' wide variety of system components.

The GIMIX CLASSY CHASSIS #19 consists of a heavyweight aluminum cabinet, constant voltage zero-voltage power supply, and SS50 Mother board with baud rate generator board... \$1398.19
Triple Disk regulator card and cables \$88.22 Baud rate generator card \$88.93
Missing cycle detector \$38.23 Filter plates \$14.92
Back panel connector plates (specify) \$8.60 50 Hz. option \$30.00

MEMORIES (GIMIX uses only Static RAM)

#67 64KB NMOS STATIC RAM board \$478.67
#64 64KB CMOS STATIC RAM board w/battery back-up \$368.64
#34 8K PROM board \$88.34
#32 16 socket PROM/ROM/RAM board \$238.32

I/O Boards (see above for Intelligent I/Os)

#41 Single port serial, RS232/20ma. current loop \$88.41
#43 2 port serial, RS232 \$126.43
#46 8 port serial, RS232 \$318.46
#42 2 port parallel \$88.42
#45 8 port parallel \$108.45
#50 serial, RS232, RS422, RS423 \$244.50
#62 SSQA serial, RS232, RS422, RS423 \$254.52
#54 ADLC serial, RS232, RS422, RS423 \$288.54
Each cable with connectors for back panel mounting (specify board) \$24.86

DISK CONTROLLERS

#68 DMA (featured in all systems above) \$588.68
#28 dbl. dens. programmed I/O (5" drives only) \$288.28
#58 single dens. programmed I/O (5" and/or 8" drives) \$278.58
#48 same as #58 but for 5" drives only \$198.48
Cable sets: 8" with Back Panel connector \$29.25
for two 8" external drives \$44.25
for two 5" drives \$34.98

SOFTWARE: GIMIX exclusive versions of OS-9/GIMIX I, II, III & FLEX are for GIMIX hardware only. All versions of OS-9 require the #68 controller. When ordered with any controller, FLEX is \$38.00

GIMIXBUG PROMs and manual \$88.66
Boot or Video boot PROM \$30.00 UNIFLEX boot PROM \$50.00
OS-9 GIMIX I \$200.00 OS-9 GIMIX II \$500.00
Editor \$125.00 Assembler \$125.00
BASIC-09 \$208.00 RUN9 \$108.00

DISK DRIVES FOR GIMIX SYSTEMS - complete with cables and power regulators.

5" DSDD 40 track 2 for \$900.00
5" DSDD 80 track 2 for \$1300.00
#88" Dual 8" DSDD drives, cabinet, power supply, & cables \$2848.86
Cabinet only \$848.18 220V 50Hz. Option, add \$38.00
Filter plate \$14.83 Cable for 2 drives \$44.82
Cable for 4 drives \$67.84 Cable for cabinet to mainframe \$46.81

WINCHESTER SUBSYSTEMS: for use only in GIMIX systems with #68

DMA controller.
#90: includes one 19MB drive, interface, and Software \$3588.90
#91: includes two 19MB drives, interface and Software \$3288.91
Contact GIMIX for price and availability of other forthcoming subsystems.

OTHER BOARDS

#78 GHOST 80X24 VIDEO BOARD \$388.78
#66 50 pin Protoboards \$56.66 #33 30 pin Protoboards \$38.33
#03 6800 CPU \$224.03
#08 6800 CPU with timers \$288.06 Baud rate option, add \$30.00
#08 RELAY DRIVER (board, bracket, transformer, and 31 relays) \$1128.08
#06 - #08 (board, bracket, transformer, without relays) \$338.08
#65 OPTO board \$348.85
WINORUSH EPROM PROGRAMMER \$376.00
3" Binder 12.00 2" Binder \$8.00

GIMIX DOES NOT GUARANTEE PERFORMANCE OF ANY GIMIX SYSTEMS, BOARDS OR SOFTWARE WHEN USED WITH OTHER MANUFACTURERS PRODUCT.

DON'T SEE IT??? ASK! OUR BROCHURE HAS MORE COMPLETE DESCRIPTIONS AND SPECS. PHONE OR WRITE TODAY FOR YOUR COPY.

BASIC-09 and OS-9 are trademarks of Microsign Systems Corp. and MOTOROLA. Inc. FLEX and UnifLEX are trademarks of Technical Systems Consultants, Inc. GIMIX, GHOST, GIMIX, CLASSY CHASSIS are trademarks of GIMIX, Inc.

GIMIX Inc.

1337 WEST 37th PLACE • CHICAGO, ILLINOIS 60609
(312) 927-5510 • TWX 910-221-4055



'68' Micro Journal

The Original FLEX™ for Color Computers

- * Upgrade to 64K
- * RS to FLEX, FLEX to RS file transfer ability
- * Create your own character set
- * Automatic recognition of single or double density and single or doubled sided
- * All features available for either single or multiple drive systems
- * Settable Disk Drive Seek Rates
- * Faster High Resolution Video Display with 5 different formats
- * Save RS Basic from RAM to Disk
- * Move RS Basic to RAM
- * Load and save function on FLEX disk
- * 24 Support Commands 12 with Source Text
- * External Terminal Program

Languages Available

Pascal, Fortran, RS Basic, RS Assembler, TSC Basic, TSC Assembler, Relocating Assembler, Macro Assembler, Mumps

If you are tired of playing games on your TRS-80C Color Computer, or find that you are handicapped by the limitations of the RS BASIC in trying to write a Program that will allow you to actually USE the Color Computer as a COMPUTER, YOU ARE READY TO MOVE UP TO THE FLEX9 Operating System. If you want to have REAL PROGRAMMING POWER, using an Extremely Powerful Business BASIC, PASCALS, C Compilers, a full-blown Macro Assembler with a Library capability so you are not continuously reinventing the wheel, YOU ARE READY TO MOVE UP TO THE FLEX9 Operating System. If you would like to see if YOU REALLY COULD USE A COMPUTER IN YOUR BUSINESS, or begin to make your Computer start PAYING ITS OWN WAY by doing some Computer Work for the millions of small businesses around you, such as Wordprocessing, Payroll, Accounting, Inventory, etc., then YOU ARE READY TO MOVE UP TO THE FLEX9 Operating System. How?? DATA-COMP has the way!

DATA-COMP's FLEX9 Conversion for the TRS-80C Color Computer was designed for the SERIOUS COMPUTER USER, with features like greatly increased Display Screens, WITH Lower Case Letters, so you can put a FULL Menu on ONE Screen, or see SEVERAL Paragraphs at the same time, with features like providing a FULL Keyboard so you have FULL Control of your Computer, AND it's Programs NATURALLY, without needing a cheat to see what Key Combination will give you what function, with USER ORIENTED functions to make using the Operating System natural, like having the Computer AUTOMATICALLY determine what type of Disk is being used in what type of Disk Drive and working accordingly, rather than you have to specify each and every thing for it, or like having the Computer work with the Printer, you have been using all along without you having to tell the new Operating System what is there, etc.

™ FLEX is a trademark of Technical Systems Consultants

DATA-COMP has everything you need to make your TRS-80C Color Computer WORK for YOU: from Parts and Pieces to Full, Ready To Use SYSTEMS. DATA-COMP designs, sells, services, and SUPPORTS Computer SYSTEMS, not just Software. CALL DATA-COMP TODAY to make your Computer WORK FOR YOU!

System Requirements

FLEX9 Special General Version x Editor & Assembler (which normally sell for \$50.00 ea.)	\$150.00
F-MATE(RS) FLEX9 Conversion Rout. for the RS Disk Controller	
when purchased with Special General FLEX9 Sys	\$49.95
when purchased without the General FLEX9 Sys	\$59.95
Set of Eight 64K RAM Chips w/ Mod. Instructions	\$59.95
Color Computer with 64K RAM and EXT BASIC	\$399.95

SPECIAL SYSTEM PACKAGES

64K Radio Shack COLOR COMPUTER, Radio Shack COLORDISK CONTROLLER, a Disk Drive System, Special General Version of FLEX9, F-MATE(RS), and a Box of 10 Double Density Diskettes: a COMPLETE, ready to run SYSTEM on your Color TV Set. \$1679.95

DISK DRIVE PACKAGES, etc.

These Packages include the Radio Shack Disk Controller, Disk Drives with Power Supply and Cabinet, and Disk Drive Cable.

PAK #1 — 1 Single Sided, Double Density Sys	\$489.95
PAK #2 — 2 Single Sided, Double Density Sys	\$749.95
PAK #3 — 1 Double Sided, Double Density Sys	\$569.95
PAK #4 — 2 Double Sided, Double Density Sys	\$919.95
PAK #5 — 2 Queue Thru Disk Double Sided Double Density Sys	\$749.95

PARTS AND PIECES

Radio Shack Disk Controller	\$179.95
1 Single Sided, Double Density Disk Drive Tandem	\$249.95
1 Double Sided, Double Density Disk Drive Queue	\$349.95
1 Queue Thru Disk Double Sided, Double Density	\$279.95
Screen Clean — Clears Up Video Distortion On Your Color Computer	\$39.95
Single Drive Cabinet with Power Supply	\$89.95
Double Drive Cabinet with Power Supply	\$109.95
Single Drive Disk Cable for RS Controller	\$24.95
Double Drive Disk Cable for RS Controller	\$34.95
Micro Tech Prods. Inc. LOWER CASE ROM Adapter	\$74.95
Radio Shack BASIC Version 1.1 ROM	\$34.95
Radio Shack Extended Basic ROM	\$69.95

TERM — AN External Terminal Driver is FREE with purchase of F-Mate MARK DATA Keyboards \$67.95

LAST OF PRODUCTION!!!

!! LIMITED QUANTITY !!

**SWTPC 8212 & 8212(W)
Intelligent Terminals**

**Data-Comp Has Bought the
Remaining Supply of
SWTPC 8212 CRT Terminals**

**If You Need an 8212
We Have Them
In Stock!**



PRINTERS

The Epson RX-80
\$375.00

The Epson MX-100
\$725.00
MX-80 \$395. FX-80 \$599.



Verbatim Diskettes.

5 Soft Sector Disks	
Single Side Single Density	\$2.75 ea
Single Side Double Density	\$2.75 ea
Double Side Double Density	\$4.92 ea
Plastic Storage Box	\$7.00 ea

8 Soft Sector Disks	
Single Side Single Density	\$3.75 ea
Single Side Double Density	\$4.10 ea
Double Side Double Density	\$6.75 ea
Plastic Library Box	\$5.00 ea

Foreign Orders Add 10% Surface—20% Air Mail



DATA-COMP

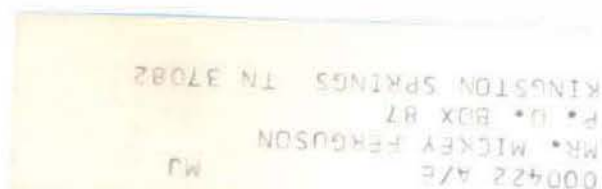
P.O. Box 794 HIXSON, TN 37343
1-615-842-4601

EM-82 Video Terminal

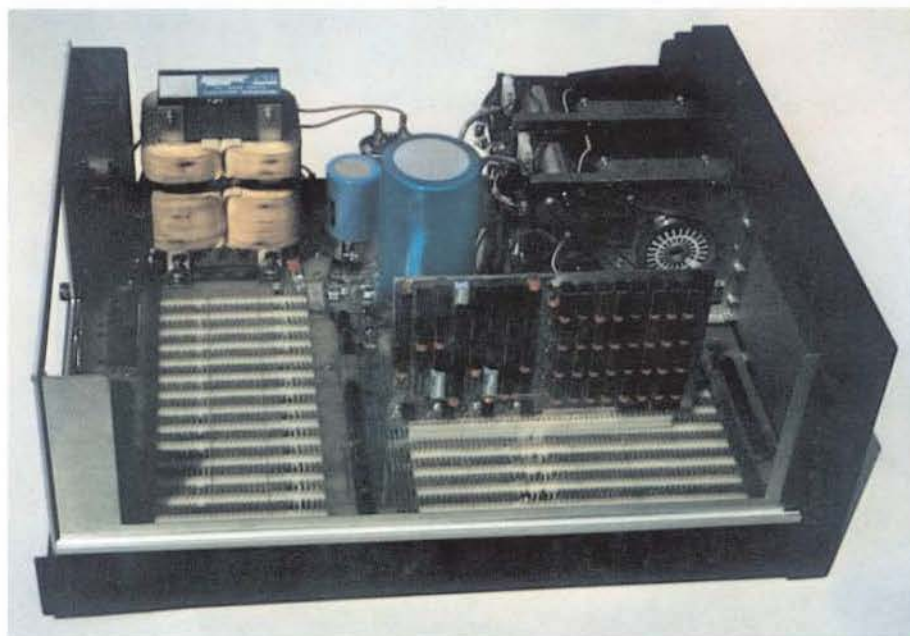
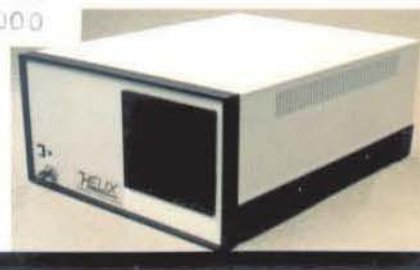
Emulates the 8200 Series from SWTPC



Data-Comp—South East Media & 68 Micro Journal Are Divisions of CPI



HELIX™



THE MAINFRAME

- Industry Standard Optima™ Cabinet
- Largest Constant Voltage Power Supply in the industry
- S-64 Bus gives 16 Bit Power and S-50 Bus Compatibility
- 10 Main (S-64) Slots
- 14 I/O (S-30) Slots plus 2 On-board
- On-board Baud Rate Generator to 38.4Kb
- Space and Power for two 5 1/4" Disk Drives
- Full Address Decoding for I/O Slots
- Two RS-232 Serial and Two parallel Ports On-board
- Single Board Construction for Reliability
- Faraday Shielded Bus Lines give "Text Book Clean" Signals

THE PROCESSORS

- 6809
- Standard 2 MHz Operation
- Standard DAT Compatible with GIMIX and SWTPC
- Standard 6840 Interval Timer
- Standard 1K Scratchpad RAM
- Standard Clock/Calendar with Battery
- Provision for Programmers Console



The HELIX™ computer system represents the latest advance in S-50 bus computer systems. Relying on the physical nature of S-50 bus connectors to guarantee compatibility, the HELIX adds 14 bus lines (becoming S-64) to allow a 68000 processor to operate with full 16 bit data transfer and 24 bit addressing, while at the same time providing full interchangeability with existing S-50 components.

Offered with a selection of processors, memories, and peripheral controllers, a HELIX system can be configured for applications ranging from advanced hobbyist to multiterminal time-sharing.

Designed to offer the utmost in speed, reliability, and utility at a reasonable price, it represents a new standard of quality for those who require a professionally designed computer for professional use.

THE POWER SUPPLY

- Ferro-resonant Transformer for Line Noise and Under-Voltage Protection
- Conservative 25 Amps at 8.5 Volts
- Conservative 5 Amps at ± 16 Volts
- Conservative Component Rating for Reliability

THE COMPONENTS

- Fully Socketed
- Gold Plated Bus Connectors
- Only "B" Series 68XX Components Used
- Only Top Grade Logic Circuits Used
- Industrial Grade Components Throughout

THE MEMORIES

- DM-64
- Field Proven
- Proprietary Memory Control Logic
- Fully Transparent Refresh
- Tested at 2.5 MHz Operation
- DM-512
- 512K Bytes on a Single S-64 Board
- 16 Bit Power and 8 Bit Compatibility
- Runs in Existing S-50 Systems where Physical Space Allows
- Full 24 Bit Addressing
- Fully Transparent Refresh

THE PRICES

Because of the variety of configurations possible, full pricing cannot be given. Representative prices are:

- 64K 6809 HELIX..... \$2495
- 256K 6809 HELIX..... \$2895
- 512K 6809 HELIX..... \$3750

HAZELWOOD COMPUTER SYSTEMS

907 E. Terra, O'Fallon, Missouri 63366 (314) 281-1055

Dealer and OEM Inquiries Invited. We support our Dealers.

Optima is a Trademark of Scientific Atlanta